

• 嵌入式系统工程 •

基于 VxWorks 的 SD 卡驱动程序的设计与实现

贺小琳^{1,2,3}, 张善从^{1,2,3}

(1. 中国科学院光电研究院, 北京 100190; 2. 中国科学院研究生院, 北京 100049;
3. 北京国科环宇空间技术有限公司, 北京 100190)

摘要: 为了提高嵌入式实时操作系统下驱动开发的可扩展性、可升级性, 研究了基于 VxWorks6.x 实时操作系统的 VxBus 虚拟总线技术。采用 XBD(extended block device)程序设计工具, 作为连接设备驱动和 VxWorks 文件系统之间的接口, 通过开发以 XBD 为接口的 SD 卡块设备驱动, 表明了 VxBus 虚拟总线技术能够有效的对设备驱动进行管理, 提高了 VxWorks 开发的性能, 并且实现了对 SD 卡的 MS-DOS FAT32 文件系统加载。

关键词: VxWorks6.x 实时操作系统; SD 卡; VxBus 虚拟总线; eXtended Block Device 扩展块设备; 驱动

中图分类号: TP368.1 **文献标识码:** A **文章编号:** 1000-7024 (2010) 16-3573-03

Development of SD card driver on VxWorks operating system

HE Xiao-lin^{1,2,3}, ZHANG Shan-cong^{1,2,3}

(1. Academy of Opto-Electronics, Chinese Academy of Sciences, Beijing 100190, China; 2. Graduate University, Chinese Academy of Sciences, Beijing 100049, China; 3. Beijing Transuniverse Space Technology Co Ltd, Beijing 100190, China)

Abstract: To extend and upgrade the driver easily on the embedded real time operating system, the VxBus virtual bus device driver infrastructure is studied. The VxBus module utilizes XBD (extended block device) library, which facilitates the interface between the device drivers and the VxWorks file system. The efficiency of VxBus technology in management of device drivers and improvement of VxWorks development is demonstrated by the realization of the SD card driver. How to apply the XBD interface is presented to the development of the SD card driver, and the MS-DOS FAT32 file system on the SD card is built.

Key words: VxWorks6.x real time operating system; SD card; VxBus virtual bus; extended block device (XBD); driver

0 引言

SD 卡是一款具有大容量、高性能、安全性好、可插拔等特点的多功能存储卡。而 VxWorks 嵌入式实时操作系统以其高实时性、高可靠性等特点在军用电子通信领域得到广泛应用。因此, 在 VxWorks 嵌入式实时操作系统下开发 SD 卡驱动具有广泛的研究与应用意义。虽然, 许多这方面的工作已经展开, 但是, 目前嵌入式操作系统的驱动开发一般通过直接修改 BSP 进行, 这种方法不利于驱动程序的模块化设计, 缺乏良好的可扩展性, 因此, 选择一种新技术代替以前的开发方法具有重要的现实意义。本文就研究了一种全新的驱动开发技术: VxBus 虚拟总线, 这种技术将各个设备挂载到虚拟总线上, 所谓虚拟总线, 就是不同于物理实现的总线, 它是按照设备功能和操作方法的不同, 将设备进行直观的易理解分类, 对每一种分类封装不同的操作接口, 提高驱动开发效率, 可扩展性和可升级性, 将开发人员从繁琐的底层代码修改中解脱出来, 将主要精力集中在提高设备性能, 完善设备功能, 为用户提供更可靠更

高效更完美的产品上。

文章阐述了在 VxWorks6.6 嵌入式操作系统下, 使用 Free-scale 的 MPC8377 处理器访问 SD 卡的 VxBus 虚拟总线存储驱动程序, 有效的提高了驱动开发的效率和性能, 实现了对 SD 卡 FAT16/FAT32 文件格式的访问, 便于在 PC 机上查看和处理 SD 卡中存储的数据文件。

1 VxWorks6.x 下 SD 卡的驱动模型

1.1 VxBus 驱动程序设计

操作系统 VxWorks6.x 版本下的驱动开发提出一个新的概念: VxBus 虚拟总线。在 VxBus 提出之前, 驱动程序是通过直接编辑修改 BSP 产生的。VxBus 提供对设备驱动的管理, 提高了 VxWork 开发的可扩展, 可升级等性能。一方面, VxBus 将各个驱动有效组织, 并将驱动对应到实际的设备上, 形成实例; 另一方面, VxBus 给上层软件提供友好的接口来调用设备, 使底层驱动对于上层用户更加透明。另外, VxBus 将设备驱动模块化, 这些模块能在 WindRiver 的开发环境 Workbench 里

收稿日期: 2009-09-29; 修订日期: 2009-11-29。

作者简介: 贺小琳 (1985 -), 女, 四川成都人, 硕士研究生, 研究方向为嵌入式系统的设计与开发; 张善从 (1976 -), 男, 福建人, 博士, 研究员, 研究方向为空间探测技术。E-mail: hexiaolin617@yahoo.cn

方便的组织和配置 ,这就意味着驱动可以加入一个工程中 ,也可以进行配置和删除 ,不需要对 BSP 进行修改 ,简化了驱动的开发过程。

按照驱动完成的不同功能 ,VxBus 将驱动分为以下几个大类 :串行驱动、存储驱动、网络接口驱动、非易失性 RAM 驱动、计时器驱动、DMA 控制器驱动、总线控制器驱动、USB 驱动、中断控制器驱动等。本文中 ,我们将 SD 卡设备按照存储驱动类进行开发。

1.2 文件系统与 XBD 工具

VxWorks 为不同类型的应用提供多种文件系统。大多数 VxWorks 文件系统依赖 eXtended Block Device(XBD)工具作为文件系统和设备驱动之间的标准接口。这种接口可以允许开发人员将文件系统和设备驱动自由组合。可移动设备的文件系统利用文件系统监督程序自动监测设备插入 ,并且在设备上实例化适当的文件系统。

SD 卡属于物理块设备 ,物理块设备是指由一系列可访问的独立数据块组成的设备。在 VxWorks 中 ,块就是指最小的可寻址单元。最常见的块设备就是磁盘。在 VxWorks 中 ,块设备的接口与其它 I/O 设备的接口不同 ,它不是与 I/O 系统直接交互 ,而是通过 eXtended Block Device(XBD)处理文件系统与块设备之间的 I/O 活动 ,它提供了文件系统与块设备之间的标准接口。

1.3 SD 卡的 XBD 驱动模型

大容量块设备驱动与文件系统进行交互 ,文件系统与 I/O 系统进行交互。XBD(extended block device)使用 ERF(event reporting framework)与驱动和高层功能交互。系统为 SD 卡设备驱动分配一个 XBD 结构。

当 SD 卡插入系统后 ,ERF 就会产生一个插入事件通知那些等待该设备插入的上层函数。XBD 工具包括两个主要数据结构 : XBD 结构 ,保存设备名 ,设备类型 ,设备方法指针 ,设备块大小以及块数等设备物理属性 ; BIO 结构 ,Block_IO (BIO) 结构包括读写 XBD 设备的必要信息 :起始块号 ,块数 ,数据传输方向和数据缓存。

SD 卡设备的 XBD 驱动模型如图 1 所示。大框中表示以下事件 :

- (1) DOS FAT32 文件系统向 ERF 登记 ,并且等待设备插入事件。
- (2) 设备驱动通知 ERF 设备插入 ,并且为 SD 卡设备创建一个 XBD 接口。
- (3) ERF 通知已经注册的 FAT32 文件系统 ,现在文件系统就可以访问 SD 卡设备了。
- (4) 文件系统在 XBD 设备驱动的上层再创建一个分区层。

另外 ,SD 卡驱动分为两层 ,靠近底层的 SD 卡主控制器驱动 Enhanced Secure Digital Host Controller (eSDHC) driver ,以及靠近上层的 XBD 大容量存储驱动 Mass Storage Class (XBD) driver。

2 SD 卡驱动程序设计

VxBus 存储驱动提供块设备管理程序 ,将各种 VxWorks 文件系统挂载到块设备上。它采用的 XBD 驱动程序设计工

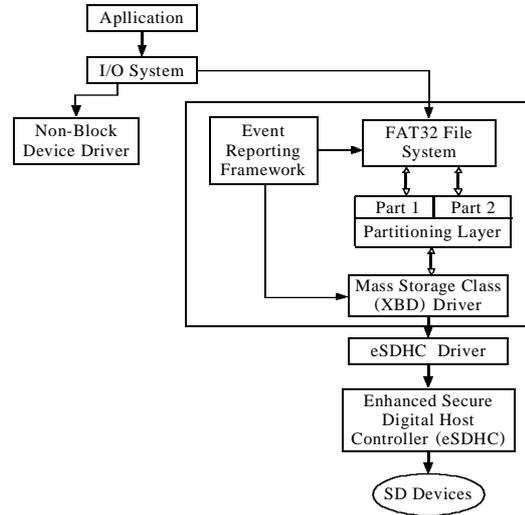


图 1 SD 卡的 XBD 驱动模型

具 ,利用两个相关的 VxWorks 子系统库 :eXtended Block Device (XBD)和 Event Reporting Framework(ERF) ,作为连接设备驱动和 VxWorks 文件系统之间的接口。驱动层次结构如图 2 所示。

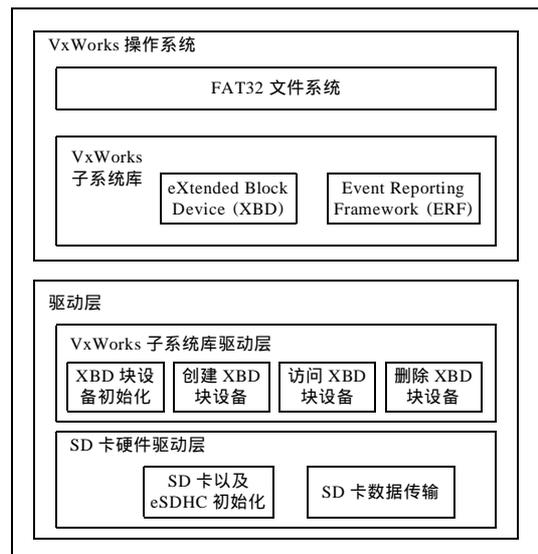


图 2 驱动层次结构

驱动层又分为两个子层 SD 卡硬件驱动层和 VxWorks 子系统库驱动层。SD 卡硬件驱动层直接操作 MPC8377 处理器上的 SD 卡控制器 Enhanced Secure Digital Host Controller (eSDHC) 中相应寄存器就可以对 SD 卡发送命令 ,完成 SD 卡初始化以及数据传输等功能。VxWorks 子系统驱动层调用 eXtended Block Device(XBD)和 Event Reporting Framework(ERF) 子系统提供的接口函数 ,完成块设备初始化 ,创建块设备 ,访问块设备以及删除块设备的功能。以下小结依次介绍驱动层的各模块实现方法。

2.1 XBD 块设备初始化

在 VxBus 驱动程序设计中 ,驱动程序首先调用 API 接口 vxbDevRegister() 通知 VxWorks 操作系统该驱动可用 ,并

且提供关于该驱动的必需信息,包括设备 ID、挂载总线类型、驱动版本、驱动名以及初始化函数集。SD 卡挂载在 PLB(processor local bus)总线上,当 PLB 驱动初始化以后,它就会发现 SD 卡直接连接在处理器本地总线上,VxWorks 系统将驱动与设备匹配在一起组成实例。此时,输入 vxBusShow 命令,就可以在注册驱动列表中找到挂载到 PLB 总线上的 eSDHC 驱动,设备实例列表中也出现了 eSDHC 实例。

驱动注册完毕以后,进行 SD 卡块设备驱动的初始化。该过程经过以下 3 个阶段: vxbEsdhcInit(), vxbEsdhcInit2() 和 vxbEsdhcConnect()。因为存储类驱动依赖 XBD 库和 VxWorks I/O 系统的其它部分,所以块设备驱动必须在它们之后进行初始化,这就意味着大部分初始化过程必须在第三阶段 vxbEsdhcConnect()进行。

在初始化的第三阶段,创建驱动程序的设备服务线程 esdhcCtrlMonitorTask,该线程程序流程如图 3 所示。主线程实现一个死循环,每循环一次检查 SD 卡控制器 eSDHC 是否初始化,然后检查卡是否刚插入,如果刚插入,将 SD 卡进行初始化,并且在 VxWorks 中创建块设备;如果卡刚拔出,将 SD 卡块设备从操作系统中删除。

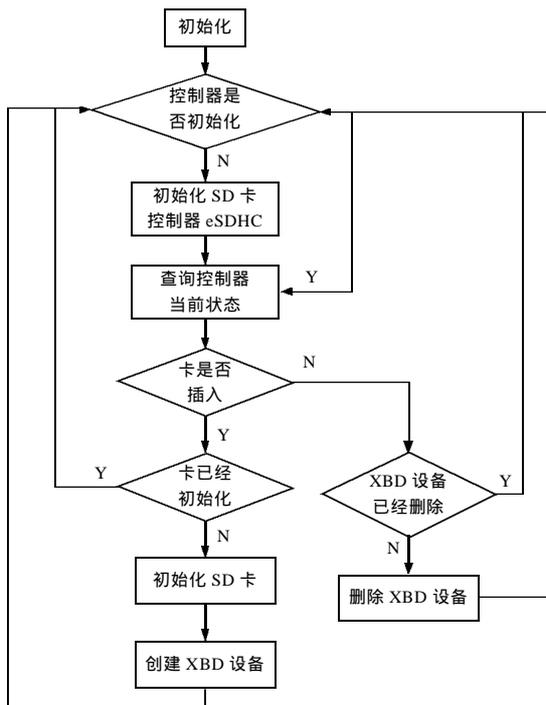


图 3 设备服务线程流程

2.2 SD 卡控制器 eSDHC 和 SD 卡初始化

SD 卡控制器 eSDHC 的初始化,包括以下内容:将系统中复用的设备 I/O 引脚配置为 SD 卡时钟线、命令线、数据线等,复位 eSDHC 控制器,打开时钟使能,配置时钟频率为 50MHZ,打开中断使能,设置大小端,地址映射模式,DMA 传输模式。

SD 卡的初始化,包括以下内容:复位 SD 卡,检查 SD 卡电压范围是否与控制器提供电压范围相符,获取 SD 卡标示号,设置 SD 卡相对地址,获取 SD 卡信息,如:容量、传输速率、时钟频率、块大小等,通过相对地址选中指定 SD 卡,设置数据

线宽度。

2.3 创建 XBD 块设备

当 SD 卡控制器 eSDHC 以及 SD 卡初始化进行完毕以后,下一步将 SD 卡作为块设备加入操作系统中。

首先创建并初始化任务同步所需信号量,然后创建执行 I/O 任务的线程 esdhcCtrlBioTask,最后通知 XBD 块设备抽象层新设备的加入。通知 XBD 块设备抽象层新设备的加入过程通过以下 3 步完成:

(1)在 SD 卡驱动中使用 ERF 子系统提供的 API 接口函数 erfHandlerRegister()向 ERF 子系统登记一个程序 esdhcXbdDevCreateSyncHandler。当 XBD 完全初始化后,ERF 系统会发出 xbdEventInstantiated 事件,程序 esdhcXbdDevCreateSyncHandler 收到这一事件后将被触发。为了避免再次被触发进而阻塞设备创建程序,由 esdhcXbdDevCreateSyncHandler 指向的函数仅仅需要调用 erfHandlerUnRegister()将自己从 ERF 中注销掉。因此,在 XBD 初始化之前,设备创建程序都不会退出。

(2)调用 API 接口 xbdAttach()向 XBD 块设备抽象层注册访问 SD 卡设备的 3 个操作函数 esdhcXbdIoctl(), esdhcXbdStrategy()和 esdhcXbdDump()。注册完毕后,xbdAttach()通过参数返回该设备的句柄,在 ERF 系统中,该句柄可以标示这一设备。

(3)调用 ERF 程序 erfEventRaise()提出 xbdEventPrimaryInsert 事件来通知上层软件系统创建了新设备。

至此,XBD 块设备创建完成。输入 devs 命令,系统设备列表中已经出现块设备 SD 卡“/sdCard”。

2.4 访问 XBD 块设备

操作系统访问 SD 卡块设备是通过上一小节提到的 3 个操作函数实现的,它们是 esdhcXbdIoctl(), esdhcXbdStrategy()和 esdhcXbdDump(),下面依次介绍这 3 个函数。

(1) esdhcXbdIoctl() 函数为多种驱动功能提供了接口,比如:设备弹出、能源管理、诊断报告、获取设备物理信息、XBD 栈初始化等。当设备弹出时,调用函数 erfEventRaise()提出 xbdEventRemove 事件来通知上层软件系统。获取设备物理信息以一张 8G 的 SD 卡为例,物理信息包括磁头数 255、每磁道扇区数 63、柱面数 996、块数 15640000、块大小 512。XBD 栈初始化完成以后,通过函数 erfEventRaise()提出 xbdEventInstantiated 事件来通知上层软件系统块设备初始化完毕,创建块设备阶段的函数 esdhcXbdDevCreateSyncHandler 收到这一事件后将被触发。

(2) esdhcXbdStrategy() 将访问存储驱动的 I/O 任务排成队列。当接到读写操作命令时,XBD 子系统将调用 esdhcXbdStrategy()函数处理文件系统传递过来的 BIO 数据结构,将这些 BIO 结构加入队列,然后释放信号量 bioReady,线程 esdhcCtrlBioTask 得到信号量 bioReady 后,解析 BIO 结构,提取出数据传输方向、传输大小、起始块号信息,将它们作为参数传递给 SD 卡硬件驱动层的数据传输模块,实现指定 I/O 任务。

(3) esdhcXbdDump(),当出现灾难性系统失败时,该程序完成最后的写数据处理。

2.5 SD 卡数据传输

SD 卡数据传输以块为单位进行,块长度为 512 字节。

(下转第 3674 页)

参考文献:

- [1] Yang S Q,Ding S L.The design and realization of CAT with cognitive diagnosis based on formal concept analysis[C].Haikou: Third International Conference on Natural Computation,2007: 241-245.
- [2] Giovanni P,Roberto P,Riccardo R.KST-based system for student tutoring[J].Applied Artificial Intelligence,2008,22(4):283-308.
- [3] 周弦,谢深泉.基于知识空间理论的自适应测试过程[J].计算机应用,2007,27(6):69-72.
- [4] 孙贝,杨贵中.基于猜测概率和失误概率的学习诊断模型[J].计

算机工程与科学,2010,32(4):154-158.

- [5] Heller J, Steiner C, Hockemeyer C, et al. Competence-based knowledge structures for personalized learning[J].International Journal on E-Learning,2006,5(3):75-88.
- [6] 林海菁,丁树良.具有认知诊断功能的计算机化自适应测验的研究与实现[J].心理学报,2007,39(4):747-753.
- [7] McGlohen M K,Wills J T,Miller G E,et al.Complementary diagnostic testing in large-scale assessment[J].Journal of Educational Measurement,Manuscript Under Review,2005,6(3):36-41.
- [8] 胡麒,何华灿.基于试题空间的学习诊断方法[J].微计算机信息,2007,23(10):238-240.

(上接第 3575 页)

根据 SD 卡协议规范,在进行读写操作之前,先发送 CMD16(SET_BLOCKLEN)命令来设置块大小。接着,CMD17(READ_SINGLE_BLOCK)命令实现单块读操作,CMD24(WRITE_BLOCK)实现单块写操作,块号作为命令参数发送给 SD 卡。

CMD18(READ_MULTIPLE_BLOCK)实现多个连续块的读操作,CMD25(WRITE_MULTIPLE_BLOCK)实现多个连续块写操作,起始块号作为参数发送给 SD 卡。多块数据传输结束后,需要向 SD 卡发送 CMD12(STOP_TRANSMISSION)命令,强制 SD 卡停止数据传输。

为了提高写 SD 卡的速度,在写操作之前加上命令 ACMD23(SET_WR_BLK_ERASE_COUNT)进行预擦除。

2.6 删除 XBD 块设备

删除各个信号量;调用 ERF 程序 erfEventRaise() 提出 xbdEventRemove 事件来通知上层软件系统该设备已经被删除;调用 xbdDetach() 在 XBD 子系统中删除该设备。

3 SD 卡设备访问

设备驱动程序建立后,对 SD 卡进行格式化 dosFsVolFormat(),应用程序就可以使用 FAT32 文件系统的基本 I/O 函数访问 SD 卡了,这些函数包括 open(),mkdir(),ll(),copy()等,方便了嵌入式 VxWorks 系统中管理数据文件。

4 实验结果与分析

使用一张 8G SD 卡作为实验设备。该 SD 卡为第二代高速卡,采用 DMA 方式,利用 4 条数据线进行数据传输,工作频率为 50MHZ。在使用文件系统的情况下,通过标准的 I/O 接口读写文件,理论最快的访问速度为 50MB/S。当文件系统 Cache 为 64KB 时,写速度为 5MB/S,读速度为 6MB/S,通过扩大文件系统 Cache 可以提高访问速度,如:文件系统 Cache 提高到 8MB,写速度为 30MB/S,读速度为 35MB/S。

5 结束语

本文介绍了在 VxWorks6.6 操作系统下,使用 VxBus 驱动模型,建立 SD 卡块设备驱动的方法。通过 VxWorks 子系统驱动层,在系统中注册该驱动,创建块设备,并且在 SD 卡上实现

FAT32 文件系统。通过 SD 卡硬件驱动层,实现 SD 卡控制器 eSDHC 和 SD 卡初始化,以及 SD 卡数据传输操作。在实际工作中已经通过调试和验证,能够很好的存储数据文件,并从 SD 卡加载应用程序执行。由于系统的低功耗、高性能、高可靠性,而且 FAT32 文件系统格式更扩展了大容量 SD 卡的支持,可以插入不同容量的 SD 卡适应不同的需要,在嵌入式系统中具有很好的应用前景。

通过使用 VxWorks6.6 下的虚拟总线 VxBus 技术,提高了编写设备驱动的效率,增加了驱动开发的可扩展性和可升级性。在 SD 卡的驱动实现中,将 SD 卡视为块设备挂载到虚拟总线上,利用 eXtended block device (XBD) 工具下的 eXtended Block Device(XBD)和 Event Reporting Framework(ERF)子系统库,作为连接设备驱动和 VxWorks 文件系统之间的接口,开发 SD 卡驱动程序,在 SD 卡上实现了 FAT32 文件系统。

参考文献:

- [1] WindRiver.VxWorks device driver developers guide[M].USA: WindRiver,2007.
- [2] Freescale. MPC8379 Reference manual [M]. USA: Freescale, 2007.
- [3] Version 2.00.SD Physical Layer Simplified Specification[S].
- [4] Version 2.00.SD Host Controller Simplified Specification[S].
- [5] 柳瑞峰,张有光.VxWorks 操作系统在数据链系统中的应用[J].计算机工程与设计,2009,30(16):3717-3719.
- [6] 刘峰,安军社.基于 VxWorks 的 1553B 仿真开发系统[J].计算机工程与设计,2006,27(24):4739-4742.
- [7] 周雪峰.实时操作系统 VxWorks 下驱动程序的设计[J].现代雷达,2007,29(1):44-45.
- [8] 王立磊.基于 VxWorks 环境与仿真终端的串口通信设计[J].计算机工程与设计,2006,27(15):2808-2811.
- [9] 魏赫颖,钱亮,贾润.VxWorks 操作系统下 CF 卡驱动程序的开发[J].微计算机信息,2008,24(3-2):64-65.
- [10] 师黎,欧阳玲,宋克.基于 VxWorks 的 POS 接口驱动软件设计[J].微计算机信息,2008,24(10-2):65-66.
- [11] 苏玉强,刘颖,张磊.VxWorks 下基于内存访问的双机通信设计[J].舰船电子工程,2008,28(4):150-153.