

文章编号:1001-9081(2005)03-0626-02

Nucleus 到 Linux 的程序移植方案

朱智勇, 黄建

(华中科技大学 电子与信息工程系, 湖北 武汉 430074)

(zhiyongwh@yahoo.com.cn)

摘要:以嵌入式环境中, MIPS 架构下, 程序从 Nucleus 到 Linux 操作系统的移植为例, 探讨了从单一地址空间操作系统到多地址空间操作系统的程序移植方案, 对多种方案进行了比较择优, 提出了单一设备、多个模块、架构整体搬迁的移植方案。从移植的结果来看, 方案可行、廉价、高效。

关键词:单一地址空间; 多地址空间; Nucleus; Linux; 移植

中图分类号: TP316.89 **文献标识码:** A

Program porting from Nucleus to Linux

ZHU Zhi-yong, HUANG Jian

(Electronics and Information Engineering Department, Huazhong University of Science and Technology, Wuhan Hubei 430074, China)

Abstract: The solutions for porting programs from single-address space operating system to multiple-address space operating system were illustrated and contrasted in the architecture of embedded MIPS. Then a solution named "single device, multiple modules and architecture porting" was introduced to the program porting from Nucleus to Linux. Experiment results show that this solution is executable, low costing and effective.

Key words: single address space; multiple address space; Nucleus; Linux; porting

0 引言

Nucleus 是单一地址空间操作系统的一种, 作为商业化的嵌入式操作系统产品, 曾被广泛使用。在 MIPS 架构中, 其操作系统和用户程序完全工作在内核模式, 且只占用和访问 0x80000000 以上的线性地址空间。因此, 在 Nucleus 中, 操作系统和用户程序工作在线性地址空间中, 且用户程序与内核服务之间没有明显的区分, 进入内核服务更像是调用 API (Application Program Interface) 而不需要上下文切换。其优点是限制少, 编程方便, 但系统健壮性差。

Linux 操作系统因其开放性和稳定性等优点, 近年来为越来越多的嵌入式设计方案所采用。它有着严格的内核模式和用户模式的区别, 在 MIPS 架构中, 用户模式只能访问 0x80000000 以下的空间, 内核模式可以访问所有的空间, 而在任何模式下访问 0x80000000 以下的空间时, 都经由 TLB (Translation Lookaside Buffer) 进行虚拟专有地址到物理地址的映射。因此, 各用户进程运行在各自虚拟地址空间内, 而非线性地址空间, 用户进程在进入内核服务时, 将以软中断的方式进行并伴随着上下文切换。其优点是系统稳定健壮, 但系统设计需要遵守特定的约束。

光纤环形网监控计费系统 PMON (Packet Over SONET Monitor) 是华中科技大学电信系与美国 Combrio 公司合作的项目。系统完成 OC48 光纤环形网上的数据抓取并转发至 12 个千兆快速以太网口, 支持基于规则的流分类、负载均衡和 NETFOLW 计费。PMON 的软件架构在 MIPS 下的 Nucleus 操作系统中已成功实现, 现将 PMON 的软件架构由 Nucleus 移植到 Linux 中, 便要面临从单一模式 (内核模式), 单一地址空

间到多模式 (内核模式和用户模式), 多地址空间的问题, 本文就此提出了一种高效、廉价的方案。

1 PMON 在 Nucleus 中的设计实现

图 1 描述了 PMON 软件架构在 Nucleus 操作系统中的实现, 其中主要包括各硬件驱动程序, 硬件驱动层的一个统一接口, 一个负责各驱动程序初始化、配置及一致性检查的模块, 一个中断服务接口, 各应用程序或进程及操作系统本身。

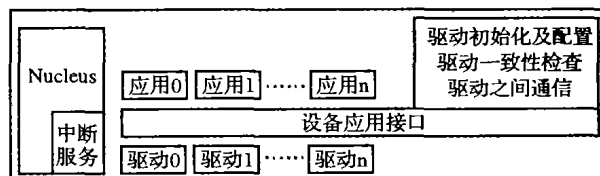


图 1 Nucleus 中的程序架构

各模块的划分只是程序在逻辑上的分割, 它们都处于同一线性地址空间中, 可以视作一个二进制的程序块, 将这个架构及程序移植到多地址空间的 Linux 操作系统中时, 便需要考虑各模块应工作在什么地址空间及什么工作模式下。

2 PMON 在 Linux 中的传统解决方案

2.1 Linux 中的传统程序架构

图 2 描述了在 Linux 操作系统中 PMON 软件的传统设计架构。由于接口已被 Linux 操作系统所规范, 程序的设计工作集中在驱动模块, 负责各驱动程序初始化、配置及一致性检查的模块, 及各应用程序。

各驱动程序处于 Linux 的内核层, 各自对 Linux 的中断管理模块申请中断。各驱动程序直接挂载于 Linux 的设备管理

收稿日期: 2004-08-30; 修订日期: 2004-10-21

作者简介: 朱智勇 (1980-), 男, 江苏无锡人, 硕士研究生, 主要研究方向: 嵌入式操作系统、宽带网络通信技术; 黄建 (1978-), 男, 四川南充人, 博士研究生, 主要研究方向: 缓存技术、主动网络技术、路由交换技术、ASIC 设计与验证。

模块,从而通过Linux的文件系统对用户层提供各自的驱动管理、应用接口,驱动的配置和一致性检查等模块将置于用户层中,同处于用户层的还有各用户进程。

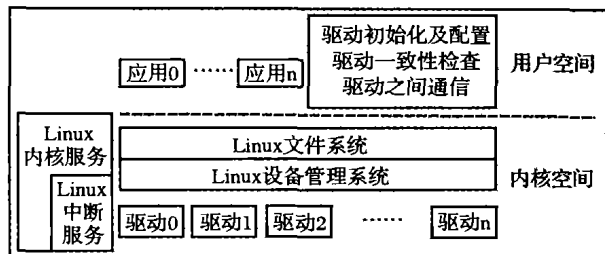


图2 Linux中的传统程序架构

2.2 性能分析

以上描述的程序设计,符合Linux架构下设计的一般原则,结构清晰。驱动程序挂载在Linux的设备管理模块上,可以利用Linux的Module特性,动态加载和卸载驱动,这对于支持热插拔的系统非常有利。

但在移植的角度看来,各驱动程序都需要为了适应Linux的接口而进行相当程度的改写,没有充分利用原来的代码而增加了工作量;各驱动程序都要直接向Linux的中断管理模块申请中断,在一个具有规模的系统中,将导致中断资源的短缺;由于各驱动都直接向Linux设备管理模块挂载,致使在文件系统中具有各自的接口,迫使对于驱动程序配置和一致性检查管理的模块置于用户层,每一次对于设备及驱动的完整及一致性检查都要进行上下文切换,效率极低。在用户进程看来,它需要面对的由文件系统提供的设备接口也比较繁杂,没有一致性的接口,调用各设备接口时,也没有底层模块为其调用设备组合的合法性作出检查和保证。

3 改进后的PMON在Linux中的解决方案

从以上分析可见,如果遵循传统的Linux程序设计来将PMON软件架构由Nucleus移植到Linux中,效果不能令人满意。为此提出一个设备多个模块,架构“整体搬迁”的方案。

3.1 改进后的PMON软件在Linux中的程序架构

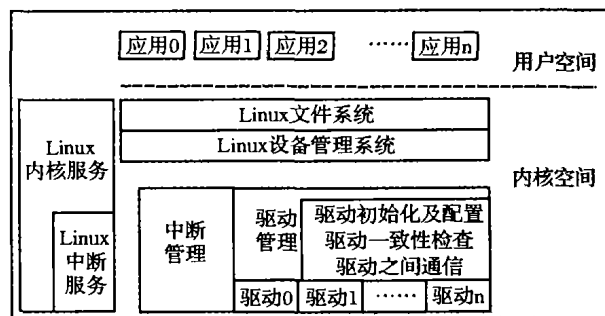


图3 改进后的程序架构

改进后的Linux中PMON程序架构如图3所示。各驱动程序、驱动管理模块(Driver Management)及中断管理模块在Linux内核中运行,而用户进程在用户空间中运行。各驱动程序编译为多个模块,而由驱动管理模块向Linux设备管理模块申请为一虚拟设备。就内核中的模块而言,实现了从单一地址空间到多地址空间的“整体搬迁”。

3.2 设计解析

改进后的PMON设计架构,摒弃了将驱动程序挂载在Linux设备管理模块的传统思维,将各驱动程序挂载在自我编写的驱动管理模块上,从而避免了为适应Linux设备管理模块,而大量改写各驱动程序的工作量。

各驱动程序没有挂载在Linux设备管理模块上,并不等于失去了Linux动态管理模块的功能,各驱动程序可编译为模块,而由驱动管理模块通过request_module()和remove_module()的内核符号调用来实现驱动的动态加载和移出主存,在嵌入式系统中有效地控制了内存资源的使用。

通过在内核中,驱动管理模块内实现驱动的配置、初始化,设备和驱动的一致性检查和驱动间的通信,避免了用户层空间和内核空间的反复陷入和上下文切换,提高了系统性能,这在一个支持热插拔的系统中,对于硬件反复检测的要求,尤为有效。

由于在Linux操作系统看来,只存在一个虚拟设备,因此,在文件系统中,向用户空间提供的接口必然统一,用户进程不必考虑设备和驱动的完整一致性,使用户进程的设计简单干净。

驱动管理模块作为一个虚拟设备,只向Linux的中断管理模块申请一个中断号,而由新架构中的中断服务模块,负责提供接口给各驱动程序用来注册中断处理函数,并负责将中断信号分发至各处理函数。这不仅满足了各驱动程序的中断要求,也解决了Linux系统中中断资源有限的问题。

新的架构的设计,将工作量主要集中到了驱动管理模块的设计上,驱动管理模块其实就是独立于Linux设备管理模块的一个简易设备管理模块,要完成以下工作:实现驱动动态加载的功能,为各驱动程序提供注册的统一接口,实现各驱动程序的配置和初始化,负责检查各设备和驱动、各设备之间和各驱动之间的合法性和一致性,负责向Linux设备管理模块注册一个虚拟设备,负责将从用户层进入虚拟设备的命令解析并分发至各驱动程序。而以上列举的驱动管理模块应完成的功能,在Nucleus中,实现各驱动程序的配置和初始化,负责检查各设备和驱动、各设备之间和各驱动之间的合法性和一致性等功能是已经完成的。

3.3 性能分析及移植结果

改进后的PMON软件在Linux中的程序设计与传统的Linux程序设计相比:1)由于最大限度地避免了用户空间陷入内核空间及内核空间返回用户空间时的上下文切换,性能显著提高。2)底层驱动为用户空间提供了统一的驱动接口,简化了用户进程的设计。各驱动程序的移植也由于避开了为Linux设备管理接口而进行的改写,从而显著提高了开发效率。3)中断服务模块的设计,有效地扩展了中断资源。4)对于Linux动态加载模块功能的有效利用,有效节约了嵌入式系统中有限的内存。

在PMON项目中,应用了改进后的程序设计,以6个人月的工作量便完成了原系统软件的移植工作,且系统性能完全满足设计要求。

4 结语

本文提出的将PMON软件架构由Nucleus移植到Linux中的方法,即单一设备、多个模块、架构整体搬迁,有效地提高了移植的效率和移植程序的性能。对于其他将单一地址空间程序移植到多地址空间操作系统的项目,也有一定的参考价值。由PMON软件架构在MIPS架构下,从Nucleus操作系统中,移植到Linux操作系统后的结果来看,方案可行、廉价、高效。

参考文献:

- [1] SWEETMAN D. See MIPS Run (1st Edition) [M]. Morgan Kaufmann, 1999.
- [2] RUBINI A, CORBET J. Linux Device Drivers (2nd Edition) [M]. O'Reilly Press, 2001.