文章编号: 1001- 2486(2009) 02- 0076- 05

麒麟操作系统层次式内核设计技术*

吴庆波, 戴华东, 吴泉源
(国防科技大学 计算机学院, 湖南 长沙 410073)

摘 要: 麒麟操作系统是高性能、高安全的国产服务器操作系统, 自主设计了层次式内核结构, 由基本内核层和系统服务层组成。基本内核层负责硬件初始化, 并提供基本的存储管理和任务管理, 系统服务层基于 FreeBSD 改进, 提供 UFS2 文件系统和 BSD 的网络协议。详细阐述了麒麟操作系统层次式内核的结构, 然后采用标准的 Benchmark 对麒麟操作系统进行了基本性能测试, 测试结果表明层次式内核结构的麒麟操作系统与宏内核结构的 UNIX 类操作系统性能相当, 最后探讨了麒麟操作系统层次式内核结构的特点和下一步发展思路。

关键词: 操作系统; 微内核; 层次式模型; 内核设计
中图分类号: TP316 文献标识码: A

The Design of Kylin Operating System's Hierarchical Kernel Structure

WU Qing-bo, DAI Hua-dong, WU Quan-yuan
(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: Kylin is a server operating system focusing on high performance and security. In this paper, a hierarchical kernel structure for Kylin operating system is proposed. Under this structure, Kylin is organized into two layers. The basic kernel layer is responsible for initializing the hardware and providing basic memory management and task management while the system service layer is based on FreeBSD providing UFS2 file system and BSD network protocols. In terms of this conception, the motivation for this novel hierarchical operating system kernel model is discussed. Then the kernel's infrastructure is introduced. Last, the performance comparison of Kylin, Redhat 9.0 and FreeBSD 5.3 with standard benchmarks is presented. Finally, a discussion of the future directions of Kylin operating system is made.

Key words: operating system; microkernel; hierarchical model; kernel design

近年来随着多核 CPU[1]、新型 I/O 体系结构和网络化应用不断涌现, 计算机硬件系统的功能越来越强大, 操作系统的复杂性也随之增加, 庞大的硬件资源, 复杂的软件层次, 使得操作系统面临复杂性控制的巨大挑战, 迫切需要研制高性能、高安全的服务器操作系统。操作系统内核结构一直是各大厂商的研究重点, 如微软公司推出了 Windows Server 2008, Google 公司针对互联网的云计算环境[2] 等。

麒麟操作系统是国家 863 计划的研究成果, 它采用层次式内核结构, 可有效满足网络环境下服务器操作系统高可扩展、高性能和高安全的需求。本文主要从操作系统内核结构设计的角度分析麒麟操作系统的性能和可扩展性。

1 相关工作

操作系统内核作为系统的基础, 其结构对操作系统的性能、安全性和可扩展性有着直接的影响, 多年来内核结构一直是操作系统领域的研究热点。目前操作系统内核结构主要有两类: 宏内核结构(Monolithic Kernel) 和微内核结构(Micro Kernel)。

以传统 UNIX 和 Linux[3] 为代表的宏内核结构操作系统, 几乎包括了操作系统的所有功能, 如任务管

* 收稿日期: 2009- 02- 10
基金项目: 国家 863 高技术资助项目(2007AA01Z177); 国家自然科学基金资助项目(90718040)
作者简介: 吴庆波(1969—), 男, 研究员, 硕士。

理、存储管理、文件系统、网络和设备驱动等, 所有功能模块共享同一地址空间。目前 Linux 的内核已发展到 2.6 版本, 并在 NUMA 支持、多线程支持[4]、多处理器调度[5] 等方面具有鲜明特色。

以 Mach[6] 为代表的微内核操作系统只在内核中提供最基本的功能, 如存储管理、任务管理和任务间通信等, 其它系统功能都由运行在用户空间中的服务程序提供, 但系统的效率较低。上世纪 90 年代, Jochen Liedtke 认为 Mach 的低效主要是设计不好, 将太多的功能包括在内核中。因此, L4[7] 从零开始设计, 内核非常小并且接口简单, L4 被称为第二代微内核操作系统。目前澳大利亚新南威尔士大学正在从事 seL4 的形式化证明工作。

K42[8] 是 IBM 发起的操作系统研究项目, 旨在提供一个新型操作系统的研究框架, 用于分析操作系统的可扩展性、性能和可管理性, 并与美国的高端计算计划有机结合, 已成为美国 HPCS(High Productivity Computing System) 项目高性能计算机的操作系统原型。

Exokernel 是麻省理工学院开发的基于微内核的库操作系统, 它的核心观点是内核支持一个最小的、高度优化的原语集, 而不提供对系统资源的抽象, 库操作系统和驱动程序以较低的优先级运行在核外。目前他们正在研究支持多核 CPU 的 Corey[9] 原型系统。

中国科技大学 MiniCore[10] 项目提出了服务体模型的新型操作系统, 该模型将存储抽象与运行抽象相分离, 采用一种新的基于消息推动的通信机制, 易于扩展到分布式计算平台。

综上所述, 宏内核结构的优点是系统功能强大、运行效率高, 但可扩展性不足, 不适于分布式计算环境和网络计算环境; 微内核结构具有灵活性强、可扩展性好等优点, 但由于使用中频繁的用户/内核态空间切换, 系统运行效率较低。因此, 针对网络化的发展趋势, 迫切需要研制一种具有可扩展性、高性能、高安全的操作系统内核结构。

2 层次式内核结构设计

麒麟操作系统借鉴了 Linux、FreeBSD[11]、Mach 和 K42 等操作系统内核技术, 结合高性能计算、网络服务、安全应用的需求, 兼顾宏内核和微内核结构的优点, 充分利用当今 CPU 多态的支持, 自主设计了层次式内核结构, 该结构由基本内核层和系统服务层组成, 如图 1 所示。

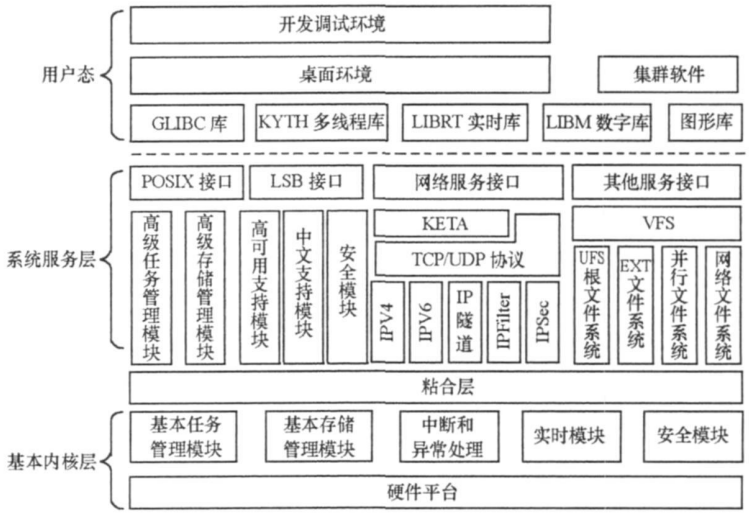


图 1 麒麟操作系统总体结构

Fig. 1 The structure of Kylin operating system

基本内核层包括硬件初始化、基本任务管理、基本存储管理、中断与异常处理等。基本内核层向下提供对硬件平台的抽象管理, 向上为系统服务层提供任务管理、中断处理、存储管理等功能。基本内核层采用模块化设计, 具有结构清晰、模块间依赖关系较弱、代码精简等特点, 便于操作系统内核的维护和移植。

系统服务层基于 FreeBSD 进行改进和优化, 为用户提供工业标准的网络、文件系统等服务接口, 实现了 Linux 二进制兼容模块、高可用模块和各种内核安全机制等, 充分利用 BSD 操作系统的稳定性和丰富的工业标准接口。

核外工具环境基于 Linux 开发, 采用 X-Window 作为系统的基本图形环境, 支持 Gnome 或 KDE 桌面环境, 设计 Windows 风格的桌面环境和控制面板, 提供简单友好的安装界面, 支持基于 B/S 结构的图形化管理工具。

与传统的微内核结构不同, 麒麟操作系统的基本内核层运行在 0 态, 系统服务层运行在 1 态, 核外工具环境运行在 3 态, 这种新的三态内核结构充分利用了 CPU 的保护技术, 保证了系统的安全性, 提高

了系统的性能。

2.1 基本任务管理

基本任务管理模块主要由基本任务调度、时钟和定时器、临界资源管理等部分组成,向系统服务层提供管理接口,用于创建、销毁、显示任务,以及设置调度参数。其中,基本任务调度子模块主要包括资源预约、调度中断、调度触发、任务派发和上下文切换等功能,时钟和定时器子模块主要提供系统时钟、调度事件的生成与定时响应机制,临界资源管理子模块支持多任务间竞争使用临界资源,提供自旋锁、互斥量、条件变量和信号量机制,如图 2 所示。

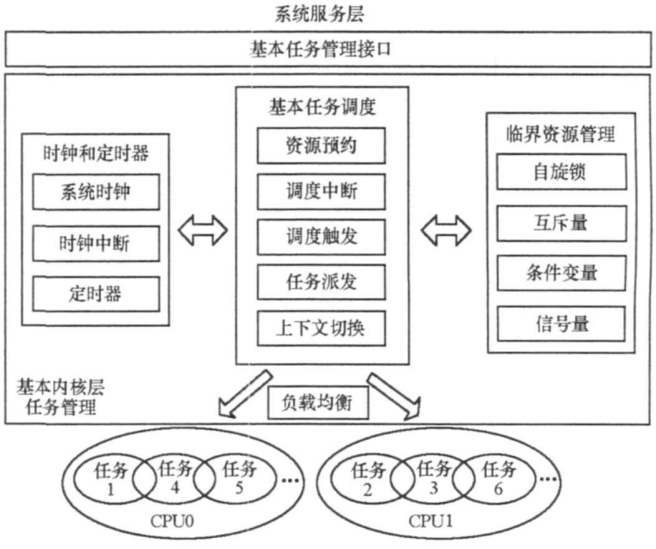


图 2 基本任务管理结构

Fig. 2 The structure of basic task management

2.2 中断处理

基本内核层捕捉和接收所有的硬件中断,并根据中断处理要求,发送到系统服务层或者基本内核的具体任务中。基本内核层支持对中断描述符表的管理,该表用于定义每个中断所对应的处理函数入口指针。为支持高效的中断转发机制,基本内核层中引入优化的事件机制,通过事件机制进行中断转发。基本内核层为中断等提供基本事件,以及 SMP IPI 事件等支持。中断与异常处理机制如图 3 所示。

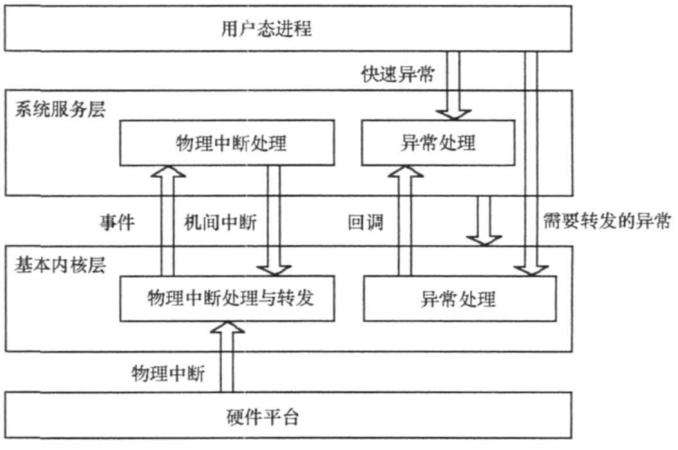


图 3 中断和异常处理机制

Fig. 3 The Mechanism of Interrupt and Fault Exception

可以看出,中断和异常处理机制相对独立。其中,中断处理采用基于事件通道的机制,当有硬件物理中断产生时,基本内核层根据系统服务层事先注册的事件处理总入口地址进行中断转发,交由系统服务层进行处理;对于异常,基本内核判断其类型,对于系统调用等需要快速处理的异常,直接写中断描述符表,其它异常利用回调函数转发给系统服务层进行处理。中断和异常处理具有下述特点:

(1) 高效的轻量级事件机制。中断被抽象成了通过事件通道传递的异步消息,通过设置回调函数,可以将中断映射到操作系统标准的中断处理机制中去,内核负责确定由谁来处理某个具体的中断。

(2) 中断和异常转发。对于系统服务层产生的中断或异常,要将其中断描述符表在基本内核中进行注册记录;当系统检测到中断或异常时,基本内核层根据中断或者异常的类型,回调系统服务层,进行中断或异常转发,并在基本内核层堆栈中创建一个中断现场,将控制交给相应的已注册过的中断或异常处理程序。

(3) 快速自陷处理。基本内核层提供快速中断(自陷)机制,基本内核层为系统调用提供一个快速句柄,该句柄可以直接由系统服务层使用,其处理不经过基本内核层。

2.3 基本存储管理

与传统的微内核操作系统不同,麒麟操作系统的基本内核层与系统服务层共享地址空间,基本内核层为系统服务层的存储管理提供支持,负责机器内存和虚存空间的分配以及内存虚实映射关系的修改,基本存储管理具有以下特点:

(1) 基本内核层和系统服务层共享同一地址空间;

(2) 不支持用户级的 Pager, 确保存储管理的高效;

(3) 提供一个全局的物理地址到线性地址的映射;

(4) 提供页表更新机制, 系统服务层对页表只有读属性, 没有写属性, 所有页表的更新由基本内核层实现。支持批处理写页表, 提高页表更新的性能。

基本内核层的代码和数据在机器启动和运行时位于物理内存的低端。基本内核层将剩余的内存划分为公共内存、基本内核的堆空间、基本内核管理数据空间和系统服务层堆空间四部分。

(1) 公共内存。这部分内存被用来存放 BIOS 信息、显示信息等, 虽然基本内核允许系统服务层访问这部分内存空间, 但公共内存不属于任何一个系统服务层。

(2) 基本内核堆空间。从基本内核的数据结束处开始, 直到基本内核预定的机器内存处结束, 均为基本内核的堆空间。在系统运行过程中, 所有非直接分配给系统服务层的内存页面均来自基本内核堆空间。

(3) 基本内核管理数据空间。从基本内核预定的机器内存开始, 为基本内核的管理数据空间。其前半部分为全局的机器内存地址到物理内存地址的映射表, 然后为页面信息表, 登记页面的管理信息。

(4) 系统服务层堆空间。基本内核管理数据空间之后的内存空间为系统服务层堆空间。基本内核从此空间中分配内存, 创建系统服务层。

2.4 系统服务层

系统服务层主要由高级进程管理、高级存储管理、网络协议、文件系统等部分组成, 它充分利用已有的开源软件进行改进和优化, 较之传统的微内核结构具有如下特点:

(1) 系统服务层性能高。与传统微内核结构不同, 系统服务层与基本内核层运行在同一地址空间, 可以通过轻量级的事件机制, 实现高效的系统服务。

(2) 硬件适配性好。传统的微内核结构驱动程序的开发有非常大的工作量, 基本上由开发人员独自完成, 支持的硬件种类有限, 已经不能适应需求。采用系统服务层结构, 可以有效结合开源社区中大量的驱动程序, 提供丰富的驱动支持。

3 测试与分析

麒麟操作系统已经取得一定的研究成果, 但还在继续完善和发展中。本节采用 LmBench[12] 测试软件, 对麒麟操作系统进行了基本性能测试。LmBench 是开源社区里面非常有影响的测试系统的综合性能软件包, 包括整型浮点运算时间、系统调用开销、文件系统和虚存、系统带宽、上下文切换时间等。测试用机配置为: Xeon 1.7G x2/1GDDR/36GSCSI/百兆网卡。

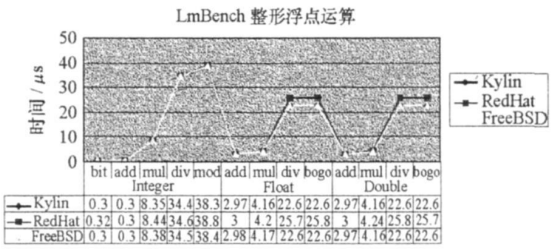
可以看出, 与传统的宏内核结构相比, 麒麟操作系统设计了精简的基本内核层和快速的中断转发机制, 并且与系统服务层共享同一地址空间, 因此在整型浮点运算时间和系统带宽方面, 麒麟操作系统与宏内核结构的系统性能相当; 在系统调用开销、文件系统和虚存延迟方面, Kylin 2.0 与 FreeBSD 5.3 基本相当, 但与 RedHat 9.0 仍有差距, 主要原因是: (1) Kylin 2.0 和 FreeBSD 5.3 采用 UFS2 文件系统, 而 RedHat 9.0 采用 Ext3 文件系统。(2) Kylin 2.0 没有直接采用 X86 体系结构提供的 SYSENTER/SYSEXIT 指令优化系统调用接口。针对 LmBench 测试结果, 我们将进一步优化文件系统性能和系统调用接口。

4 总结及下一步工作

麒麟层次式内核结构设计兼顾了宏内核和微内核结构的优点, 具有可扩展性好、安全性强、高可用等优点。层次式内核结构与硬件相关部分是在基本内核层实现的, 并采用模块化设计思想, 便于移植到新的硬件平台。与微内核操作系统不同, 麒麟的基本内核层提供了较丰富的接口原语, 可根据需要灵活支持系统服务层, 并可与未来的可信硬件平台[13] 有机结合, 形成自主的可信平台。目前国际上正基于微内核技术或虚拟机技术构建可信的操作系统平台, 如 Stanford 大学的 Terra[14], 微软的 NGSCB[15] 等。

随着网络时代服务器“集约化”的发展趋势, 课题组将基于麒麟特有的层次式内核结构, 在动态资源管理、系统高可用、高安全等方面开展工作, 力争早日实现具有自己特色的高可信网络化服务器操作系

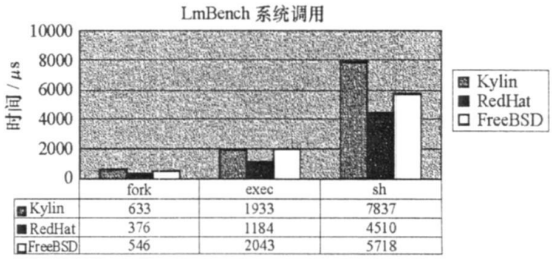
统平台。



	bit	add	mul	div	mod	add	mul	div	bogo	add	mul	div	bogo
	Integer					Float				Double			
Kylin	0.3	0.3	8.35	34.4	38.3	2.97	4.16	22.6	22.6	2.97	4.16	22.6	22.6
RedHat	0.32	0.3	8.44	34.6	38.8	3	4.2	25.7	25.8	3	4.24	25.8	25.7
FreeBSD	0.3	0.3	8.38	34.5	38.4	2.98	4.17	22.6	22.6	2.97	4.16	22.6	22.6

图 4(a)　整型浮点运算时间

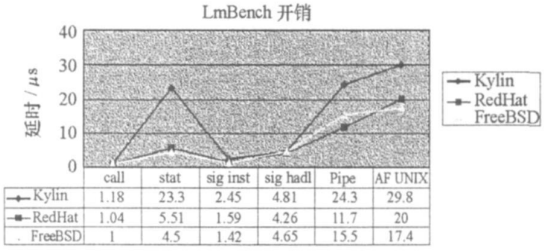
Fig. 4(a)　Simple arithmetic operations/ second



	fork	exec	sh
Kylin	633	1933	7837
RedHat	376	1184	4510
FreeBSD	546	2043	5718

图 4(b)　系统调用

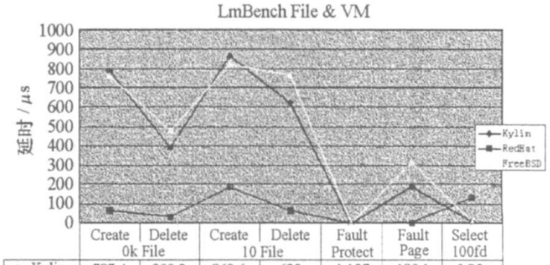
Fig. 4(b)　System call latencies



	call	stat	sig inst	sig hadl	Pipe	AF UNIX
Kylin	1.18	23.3	2.45	4.81	24.3	29.8
RedHat	1.04	5.51	1.59	4.26	11.7	20
FreeBSD	1	4.5	1.42	4.65	15.5	17.4

图 4(c)　系统开销

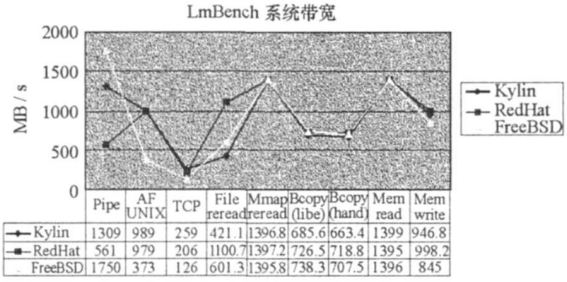
Fig. 4(c)　System call overhead



	Create 0k File	Delete 0k File	Create 10 File	Delete 10 File	Fault Protect	Fault Page	Select 100fd
Kylin	787.4	390.3	860.6	623	1.127	189.1	9.85
RedHat	62.9	34.9	185.4	64.7	1.345	4.2541	131.8
FreeBSD	759.3	474.5	821	762.8	0.767	312.2	6.69

图 4(d)　文件系统和虚存延迟

Fig. 4(d)　File and VM system latencies



	Pipe	AF UNIX	TCP	File reread	Mmap reread	Bcopy (libc)	Bcopy (hand)	Mem read	Mem write
Kylin	1309	989	259	421.1	1396.8	685.6	663.4	1399	946.8
RedHat	561	979	206	1100.7	1397.2	726.5	718.8	1395	998.2
FreeBSD	1750	373	126	601.3	1395.8	738.3	707.5	1396	845

图 4(e)　系统带宽

Fig. 4(e)　Local communication bandwidths

参 考 文 献:

[1] Fedorova A, Seltzer M, Small C, et al. Performance of Multithreaded Chip Multiprocessors and Implications for Operating System Design[C]// USENIX' 05. Anaheim, 2005: 395- 398.

[2] Ghemawat S, Gobioff H, Leung S T. The Google File System[C]// The 19th ACM Symposium on Operating System Principles, New York, 2003: 29 - 43.

[3] Bovet D P, Cesati M. Understanding the Linux kernel[M]. O' Reilly Media, 2005.

[4] Bligh M J, Dobson M, Hart D, et al. Linux on NUMA Systems[C]// The Linux Symposium. Ottawa, Canada, 2004, 1: 89- 102.

[5] Siddha S, Pallipadi V, Mallick A. Chip Multi Processing Aware Linux Kernel Scheduler[C]// The Linux Sysmposium, Ottawa, Canada, 2005, 2: 193- 203.

[6] Rashid R, Lannamico L, Dean R. The Mach Project Home Page[DB]. http://www. cs. cmu. edu/ afs/ cs/ project/ mach/ public/ www/ mach. html.

[7] Liedtke J. The L4Ka Project Home Page[DB]. http:// www. l4ka. org.

[8] Appavoo J, Auslander M, Burtico M. K42: An Open-source Linux-compatible Scalable Operating System Kernel[J]. IBM Systems Journal, 2005, 44(2): 427- 440.

[9] Silas B W, Chen H B, Chen R, et al. Corey: An Operating System for Many Cores[C]// The 8th USENIX Symposium on Operating Systems Design and Implementation, 2008: 43- 57.

[10] 李宏, 陈香兰, 吴明桥, 等. 服务体模型与操作系统内核设计技术[J]. 计算机研究与发展, 2005, 42(7): 1272- 1276.

[11] Marshall K M, George V, Neville N. The Design and Implementation of the FreeBSD Operating System[M]. Boston: Addison-wesley Professional, 2004.

[12] http:// www. bitmover. com/ lmbench/.

[13] Intel Trusted Execution Technology Architectural Overview[R]. Intel Incorporation, 2003.

[14] Garfinkel T, Pfaff B, Chow J, et al. Terra: A Virtual Machine-based Platform for Trusted Computing[C]// The 19th ACM Symposium on Operating Systems Principles, New York, 2003: 193- 206.

[15] Peinado M, Chen Y, England P, et al. NGSCB: A Trusted Open System[C]// The 9th Australasian Conference on Information Security and Privacy, Sydney, Australia: Microsoft Corporation, 2004: 86- 97.