

双模式基础平台子系统结构风格研究^{*}

Study of Structural Styles of Base-Platform Subsystems Using Double-Modes

耿 技 周明天 秦志光

(电子科技大学计算机学院 成都610054)

Abstract Base-Platform Subsystem is a kind of key component contained in general structures of many operating systems. Base-platform subsystems using double-modes have used double-mode structural style. General structures of Base-platform subsystems using double-modes are further classified in this paper. Structural styles used in key components of base-platform subsystems using double-modes are also discussed in this paper. In our opinion, it is very important and quite helpful for designers of base-platform subsystems using double-modes to systematically realize the structural knowledge of base-platform subsystems using double-modes.

Keywords Operating system, Architecture structural style, Microkernel

一、引言

基础平台子系统是大多数操作系统之总体结构所包含的一种关键部件。作者在文[1]中系统地讨论了一些常见的基础平台子系统总体结构风格。从并发结构观点看,大多数操作系统其基础平台子系统总体结构使用了双模式结构风格。若一个基础平台子系统其并发结构观点下的总体结构使用了双模式结构风格,那么我们把该基础平台子系统称为双模式基础平台子系统。本文将基于文[1]的有关结论讨论双模式基础平台子系统总体结构的进一步分类,并讨论双模式基础平台子系统其关键部件所采用的一些常见的结构风格。在我们看来,系统地了解双模式基础平台子系统的结构知识对双模式基础平台子系统设计者而言非常重要也十分有益。

二、双模式基础平台子系统的结构风格

不言而喻,双模式基础平台子系统其总体结构包含两个模式模块,它们分别在两种不同的特权模式下运行。在下面的讨论中,我们把双模式基础平台子系统的这两个模式模块分别称为核外子系统和核心子系统。同时,把核外子系统所使用的特权模式称为用户模式,把核心子系统所使用的特权模式称为核心模式,但用户模式的特权级别低于核心模式的特权级别。

图1示意性地描述了双模式基础平台子系统总体结构风格。根据核外子系统的内部语义逻辑和行为逻辑

的组成内容,可以把双模式基础平台子系统总体结构进一步分为两种类别,属于第一种类别的双模式基础平台子系统总体结构其核外子系统的内部语义逻辑和行为逻辑仅仅由应用软件的内部语义逻辑和行为逻辑组成;属于第二种类别的双模式基础平台子系统总体结构其核外子系统的内部语义逻辑和行为逻辑则由应用软件的内部语义逻辑和行为逻辑以及基础平台子系统的部分内部语义逻辑和行为逻辑共同组成。显然,在第一种类别的双模式基础平台子系统总体结构中,核心子系统是一个完整的基础平台子系统,正是因为如此,人们习惯上把双模式基础平台子系统总体结构的第一种类别称为单体内核结构风格。很明显,在第二种类别的双模式基础平台子系统总体结构中,核心子系统只是基础平台子系统的一个组成部分。换句话说,在第二种类别的双模式基础平台子系统总体结构中,基础平台子系统被分解成两个部分:其中一部分在核外子系统中实现,另外一部分则作为核心子系统实现。我们把前者称为基础平台子系统的扩展服务子系统,把后者称为基础平台子系统的基本内核子系统。在下面的讨论中,还将把双模式基础平台子系统总体结构的第二种类别称为基本内核结构风格。图2和图3分别示意性地描述了双模式基础平台子系统采用的单体内核结构风格和基本内核结构风格。由于在核心模式下开发程序较为困难,因此越来越多的多任务操作系统其基础平台子系统总体结构在选用双模式结构风格的同时选用基本内核结构风格。

^{*} 本文为信息产业部/四川省经贸委专项课题“安全操作系统”系列研究报告之一。

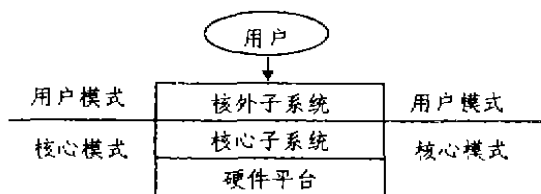
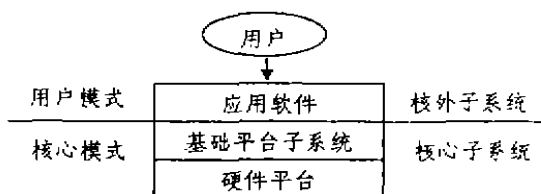
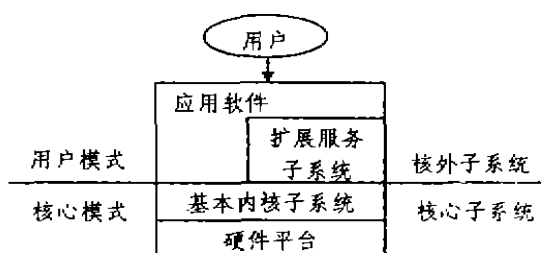


图1 双模式基础平台子系统总体结构风格示意图

图2 双模式基础平台子系统
单体内核结构风格示意图图3 双模式基础平台子系统
基本内核结构风格示意图

毫无疑问,在操作系统工程实践中,当使用基本内核结构风格构造双模式基础平台子系统总体结构时,人们必须解决如何把基础平台子系统分解为扩展服务子系统和基本内核子系统这样一个基本问题。近些年来受到人们广泛重视的微核结构设计思想为解决该问题提供了一种思路。通过总结关于微核结构设计的研究成果和实践经验^[2~11],我们可以把微核结构设计思想表述为:尽最大努力剔除核心子系统多余成份并把它移到核外子系统中实现,核心子系统只实现一些必要的简单概念及其服务从而保持核心子系统简洁高效。由于核心子系统是整个计算机系统中使用频率最高的关键部件,因此使其可靠、高效是非常必要的,不难理解,一个简洁的核心子系统将易于排错。因此,一个简洁的核心子系统将具有很高的可靠性,必须指出,微核结构设计思想并未给出核心系统的设计规范。在操作系统工程实践中,人们常常根据自己对微核结构设计思想的理解定义核心子系统的功能性需求。尽管 Mach^[5]的设计者们首先明确倡导微核结构设

计思想,但许多研究者认为 Mach 的核心子系统仍然不够精简,一些研究者提出了纳核(Nanokernel)^[12]和外核(Exokernel)^[16,17]概念。在我们看来,无论是微核、纳核还是外核,都是微核结构设计思想运用于操作系统工程实践中的产物。

下面我们分别讨论扩展服务子系统以及基本内核子系统常用的运行组织形式。这里所谓的运行组织形式实际上就是并发结构观点下的总体结构风格。用运行组织形式指代并发结构观点下的总体结构风格其目的在于简化有关内容的叙述。

2.1 扩展服务子系统的运行组织形式

进程组织形式是扩展服务子系统常用的一种运行组织形式。从原理上讲,进程组织形式可以被进一步分为两种:多进程组织形式和单进程组织形式。与单进程组织形式相比较,多进程组织形式有利于实现扩展服务子系统的可定制性、可分布性等非功能性需求,不过,由于多个扩展服务进程相互合作会导致地址空间之间的切换,因此多进程组织形式会降低扩展服务子系统的性能。与多进程组织形式不同,单进程组织形式不会影响扩展服务子系统的性能。然而,单进程组织形式不利于实现扩展服务子系统的可定制性、可分布性等非功能性需求,总的来说,在操作系统工程实践中,多进程组织形式比单进程组织形式更为常用。就多进程组织形式而言,我们可以把它进一步分为两种:嵌套进程组织形式与平铺进程组织形式。所谓嵌套进程组织形式就是文[1]中介绍过的嵌套进程结构风格。在文[1]中,我们把它视作功能结构观点下的一种结构风格。实际上,它也是并发结构观点下的一种结构风格。这种结构风格既可以用作基础平台子系统的总体结构风格,也可以用作基础平台子系统的部件结构风格。对于扩展服务子系统来说,嵌套进程组织形式目前仍处于试验阶段^[18],而平铺进程组织形式却已被广泛用于各种产品操作系统(比如 Mach^[5]、QNX^[7]以及 Windows NT^[19]等)中。

库组织形式也是扩展服务子系统常用的一种运行组织形式,我们同样可以进一步把它分为两种:动态库组织形式和静态库组织形式。这两种库组织形式之间的差异主要体现在它们分别使用两种不同的程序链接技术加以实现。具体地说,动态库组织形式使用程序的动态链接技术加以实现,静态库组织形式使用程序的静态链接技术加以实现。由于动态链接技术支持程序的局部更新,因此动态库组织形式有利于实现扩展服务子系统的可定制性。正是因为具有这一特点,动态库组织形式近些年受到了人们越来越多的关注。在现代操作系统工程实践中,人们不仅把动态库组织形式用于扩展服务子系统,还把这种库组织形式用于基本内

核于系统^[20]。

习惯上,当扩展服务于系统采用进程组织形式时,人们把相关的核外于系统的运行组织形式称为客户/服务器组织形式^[21]。在采用客户/服务器组织形式的核外于系统中,应用软件与扩展服务于系统被隔离在不同的地址空间中。因此,进程组织形式有利于实现扩展服务于系统的可靠性、安全性等。与此不同,当扩展服务于系统采用库组织形式时,在相关的核外于系统中,应用软件与扩展服务于系统同处在相同的地址空间中。因此,库组织形式不利于实现扩展服务于系统的可靠性、安全性等。显而易见,当扩展服务于系统采用库组织形式时,我们可以采取访问控制这种基本的保护措施以增强扩展服务于系统的可靠性、安全性等。保护共享库^[22]正是人们在库组织形式基础上采取访问控制这种保护措施后所获得的一种新的用于扩展服务于系统的运行组织形式。当然,保护共享库在增强扩展服务于系统的可靠性、安全性的同时也会降低扩展服务于系统的性能。不难理解,如果一个基础平台于系统其总体结构同时使用单地址空间结构风格^[1]和基本内核结构风格,那么该基础平台于系统其扩展服务于系统的运行组织形式就只能使用库组织形式^[6,11]。

2.2 基本内核子系统的运行组织形式

图4示意性地描述了基本内核于系统常用的一种运行组织形式。尽管该图是一个示意图,但它仍然清楚地表明:采用图示运行组织形式的基本内核于系统将被分解为上半部和下半部两个基本部件,它们之间的区别主要表现在三个方面:一是,它们依赖的上下文不同;二是,它们实现的功能不同;三是,激活它们的原因不同。这里所谓的上下文,简单地说,就是程序在运行过程中可以访问的各种资源所组成的集合。一般说来,上半部在应用软件或扩展服务于系统的上下文中运行,其功能主要是,为应用软件 and 扩展服务于系统提供各种服务。当应用软件或扩展服务于系统在运行过程中产生各种异常事件时,上半部也将处理这些事件,显然,当应用软件或扩展服务于系统引用上半部提供的服务时,或者当应用软件或扩展服务于系统在运行过程中产生异常事件时,上半部均将被激活。与上半部不同,下半部在专用的系统上下文中运行,它专门处理由硬件平台产生的各种中断事件。十分明显,当硬件平台产生中断事件时,下半部将被激活。顺便指出,在基本内核结构风格中,基本内核于系统与核心于系统这两个术语所指称的对象是相同的。因此,图4所描述的运行组织形式也是核心于系统的运行组织形式。这种运行组织形式同样可以用于单体内核结构风格中的核心于系统。

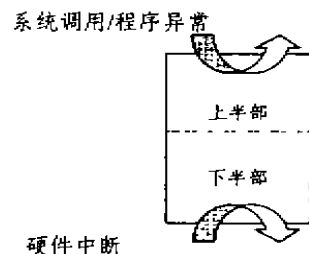


图4 一种常见的基本内核于系统运行组织形式示意图

结束语 本文系统地讨论了双模式基础平台于系统所采用的一些常见的结构风格。本文所讨论的结构风格既包括双模式基础平台于系统的总体结构风格,也包括双模式基础平台于系统中某些关键部件的结构风格。不可否认,本文所讨论的双模式基础平台于系统结构风格是不完备的,本文对双模式基础平台于系统有关结构风格的分析评价也是粗浅的。好在我们的目的只是希望,在操作系统体系结构领域本文能够产生抛砖引玉的效果。对于本文的不足乃至错误之处,敬请各位专家批评指正。

参考文献

- 耿技,周明天,秦志光 操作系统结构风格研究. 计算机科学, 2001, 28(11)
- Anderson T E. The Case for Application-Specific Operating System. In: Proc. of the Third Workshop on Workstation Operating Systems, 1992. 92~94
- Engler D R, et al. The Operating System Kernel as a Secure Programmable Machine. In: Proc. of the Sixth SIGOPS European Workshop, 1994. 62~67
- Liedtke J. On Micro-kernel Construction. In: Proc. of the Fifteenth ACM Symposium on Operating Systems Principles, 1995. 237~250
- Black D L, et al. Microkernel Operating System Architecture and Mach. In: Proc. of the USENIX Workshop on Microkernels and Other Kernel Architectures, Seattle, Washington, 1992. 11~30
- Marsh B D. Multi-Model Parallel Programming. [PhD Thesis] Department of Computer Science, University of Rochester, 1992
- Hildebrand D. An Architectural Overview of QNX. In: Proc. of the Usenix Workshop on Micro-kernels and Other Kernel Architectures, Same to[5], pp. 113~126
- Bomberger A C, et al. The KeyKOS Nanokernel Architecture. Same to [5], pp. 95~112
- Bershad B N, et al. SPIN---An Extensible Microkernel for Application-Specific Operating System Services. [Technical Report TR 94-03-03] University of Washington, Feb. 1994

(下转第112页)

具有代表性又要尽可能全面。测试对象的选择也要有足够的代表性,使其反映不同类型用户的分布。

2)测试环境的安排要尽量使测试对象轻松自然。在有的测试中,由于条件所限,测试管理人员对测试数据进行现场人工记录,而未使用低干扰性的工具如摄像机等,这可能会使测试对象感到紧张,影响测试结果的客观性。

3)对比性测试可使可用性测试效果更加显著。孤立的一种产品的测试结果容易使人感到缺乏参照,不能形成清晰的概念。而同类产品的对比性测试会清楚体现出不同产品在可用性质量方面的优劣,使得测试结果更有意义。

4)CIF测试主要是对成品的可用性质量进行评价,因此测试采取无协助的方式,但在测试中仍发现一些可以改进的软件可用性问题。因此也可以基于这种方法由测试人员进行引导式的协助性测试来发现问题,帮助改进产品设计。

总结 目前,IUSR计划正处在CIF测试在工业界的两年试用阶段的第二年,美国国家标准技术局以及主要由工业界巨头组成的IUSR协作体正在积极推进它的完善和应用,欧盟的UsabilityNet计划以及日本的有关政府和民间机构也将CIF测试列为一项重要的推广内容。据最新消息,CIF测试的V2.0版正在

修订当中,将在近期内做为美国国家信息技术委员会标准(NCTIS)颁布,并进一步成为ISO国际标准。鉴于CIF测试是一项主要由IT工业界自发形成的标准,并且由IT工业发达的国家来推动,可以相信它在今后将为工业界所广泛采用,并带来产品可用性质量的普遍提高。这值得我国的学术界和工业界及早关注并进行相应的准备。作为欧盟可用性中国支持中心(Chinese Support Center for EU UsabilityNet),我们正在与美国国家标准技术局合作在国内工业界进行CIF测试的试点和推广工作。

致谢 本文工作得到了欧盟委员会第五框架研究开发计划和中国政府科学技术部国际科技合作计划的支持。

参考文献

- 1 Bevan N. Tutorial 11--Industry Standard Usability Tests. Interact'99 Conference, Edinburgh, 1999
- 2 Nilsen J. Usability Engineering. Academic Press, 1993
- 3 Blanchard H E. Standards for Usability Testing ACM SIGCHI Bulletin, 1998, 30(3)
- 4 Bevan N. Common Industry Format Usability Tests. In: Proc. Usability Professionals Association Conf. Scottsdale, Arizona, 1996. 5
- 5 Engler D R. The Design and Implementation of a Prototype Exokernel Operating System. [MA Thesis]. MIT, Oct. 1998
- 6 Engler D R. The Exokernel Operating System Architecture. [PhD Thesis]. MIT, Oct. 1998
- 7 Ford B, et al. Microkernel Meet Recursive Virtual Machines. In: Proceedings of the Second Symposium on Operating Systems Design and Implementation, Oct. 1996. 137~152
- 8 Custer H. Inside Windows NT. Microsoft Press, Redmond, WA, 1993 (见:程渝荣译,《Windows NT 技术内幕》,清华大学出版社,1993)
- 9 陈莉君,编著. Linux 操作系统内核分析. 人民邮电出版社, 2000
- 10 Tanenbaum A S, Woddhull A S. Operating Systems: Design and Implementation. Second Edition. Prentice Hall, Englewood Cliffs, NJ, 1997
- 11 Banerji A, Cohn D L. Protected Shared Libraries. A New Approach to Application-Extensible Operating Systems. [Technical Report TR-94-35]. Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN, 1994
- 12 Seltzer M, et al. An Introduction to the Architecture of the VINO Kernel. In: VINO: The 1994 Fall Harvest. [Technical Report TR-34-94]. Harvard University, 1994
- 13 Cheriton D, Duda K. A Caching Model of Operating System Kernel Functionality. In: Proc. of the First Symposium on Operating Systems Design and Implementation, 1994. 179~193
- 14 Probert D, Bruno J. Building Fundamentally Extensible Application-Specific Operating Systems in SPACE. [Technical Report TRCS95-06]. Computer Science Department, UC Santa Barbara, Mar. 1995
- 15 Roscoe T. The Structure of a Multi-Service Operating System. [PhD Thesis]. Queens' College, University of Cambridge, Apr. 1995
- 16 Campbell R H, Tan See-Mong. μ Chioes: An Object-Oriented Multimedia Operating System. In: Proceedings of the Fifth Workshop on Hot Topics in Operating Systems, Orcas Island, Washington, May 1995
- 17 Veitch A C, Hutchinson N C. Kea---A Dynamically Extensible and Configurable Operating System Kernel. In: Proc. of the Third Conf. on Configurable Distributed Systems, 1996

(上接第121页)