一个教学用操作系统的设计与实现

王红玲 褚亚铭 吕强
（苏州大学 计算机科学与技术学院，江苏 苏州 215006）

摘 要：本文首先指出了现有的教学用操作系统存在的不足，然后设计并实现了一个运行在虚拟机上的微内核结构的教学用操作系统，描述了系统中进程管理、进程间通信、基本内存管理、磁盘服务器以及文件服务器的设计和实现。本系统的实现将有利于学生从微观上观察操作系统的行为特征，并且帮助他们理论联系实际，具有较好的教学价值。

关键词：微内核；操作系统
中图分类号：G642 文献标识码：A
文章编号：1672-5913（2007）08-0063-03

A Design and Implementation of an Operating System for Tutorial

WANG Hongling ZHU Yaming LV Qiang
(School of Computer Science and Technology, Soochow University, Suzhou, 215006, China)

Abstract: This paper firstly points out the weaknesses of current popular operating systems for tutorial, then designs and implements a microkernel operating system for tutorial on Bochs. It describes the design and implementation of process management, IPC, basic memory management, file system server and disk server in detail. The system will do benefits to students in helping them learning operating system principles and offering them a platform to practices what they have learned in class.

Keywords: microkernel; operating system

0 引言

操作系统（Operating System）是计算机系统中最重要的系统软件，是硬件的第一层封装与抽象，在计算机系统中占据着重要的地位。操作系统课程是计算机专业学生的必修重点课程，其目的在于使学生掌握操作系统的基本概念与原理并为今后的相关技术学习与研究打下良好的基础。

但是在多年的教学实践中，始终存在着教师觉得不好教，学生觉得不好学的问题。造成这个问题的重要原因之一在于这门课程的课堂理论教学环节相对比较成熟，但是实践教学环节却相对滞后，或者说两者之间的平衡把握的不好[1]。IEEE/ACM在2001年的本科生教学建议中再次强调，操作系统的教学在进行理论讲授的同时，必须结合相当数量的动手实践[2]。只有通过阅读操作系统的源代码，并且亲自动手对其进行修改与扩展即强调实践环节[3]，学生才有可能对操作系统课本所讲述的抽象原理有比较深刻的理解。因此一个合适的教学专用操作系统实验平台对本科生操作系统课程的顺利开展有着重要的意义。

1 常用实验操作系统简介

目前国内外大学在讲授操作系统课程时，主要使用的实验操作系统有Linux[4][5]、Minix[6]以及Nachos[7]。从纯粹教学的角度来看，这几个教学平台各有特点又各自存在着某些方面的不足。归纳起来，现有的用于教学的操作系统平台存在着以下三个问题：

（1）绝大多数系统安装在裸机上。系统在承担实验平台这个角色的同时还要承担起系统支撑环境的角色，这种角色上的重叠使得调试跟踪以及修改扩展都

很不方便。

(2) 过于实用，导致代码量过于庞大。这些强调实用的操作系统将学生置身于一个技巧与实现细节的大森林，在这种情况下学生更多的是挫折感和失落感，同时很难把握住其中的关键点并且走出这个大森林。

(3) 对体系结构强调的不够。前面提到的三款操作系统中除Minix以外都是传统的单一内核结构的操作系统。

这些问题的存在，影响了学生（特别是本科生）课程实验的顺利开展。

2 系统设计

针对上述存在的问题，我们设计了一个针对大学操作系统课程的实验系统。系统的设计理念是“弱化实用功能，突出实现原理”，在力求说清楚操作系统原理中一些重点与难点的大前提下，适当地对具体细节作简化，尽可能控制系统的规模，以避免一些枝节的问题影响学生对重点的理解。

2.1 总体设计

系统的设计目标是：完备性、可扩展性和可操作性。

所谓完备性是指系统应包含操作系统中的基本概念与原理，主要包括进程管理、内存管理、文件系统管理和设备管理四大模块。针对每个模块的核心概念设计相应代码，在此基础之上各个模块又可以整合成一个完整的可运行系统，从而在体现各部分实现原理的同时，向学生展示一个全貌，让他们了解各部分是如何有机地组合到一起的。

在达到内容完备性的前提之下，系统还应该力求便于学生的阅读与扩展。由于微内核操作系统同单一内核操作系统相比，具有结构更加清晰、更容易扩展等优点[8][9]，所以我们采用微内核体系结构作为系统的体系结构。

另外，为了保证系统的可操作性，系统采用了Bochs虚拟机[10]作为底层运行平台。理由是：Bochs完全模拟了i386，使得系统不失真实性。同时Bochs提供的强大的配套调试工具便于学生跟踪了解系统运行的具体状态与精确时序。由于使用了虚拟机，避免了学生在硬件故障排除这个比较棘手的问题上花费大量不必要的时间，并且大大节省了编辑、编译、下载、调试的时间周期，系统也不需要频繁地重启。

整个实验平台的逻辑结构图如图1所示。其中微内核直接运行于Bochs虚拟机之上，它在完成了对硬件的封装与抽象之后，向应用层提供了一个支撑环境，而各个系统服务器与用户进程则运行于此支撑环境之上。

图1

2.2 系统结构与模块划分

在构建整个实验平台时，系统根据功能被划分为微内核与核外服务器这两大部分。各个服务器与用户进程通过微内核这一消息总线进行消息传递，如图2所示。

图2

2.2.1 微内核部分

系统的微内核仅仅实现一些最为基本的服务，它为整个系统的正常运作提供基础保障。它被实现在核心级，可以执行特权指令，这是因为这部分实现高度依赖底层的硬件。微内核直接与底层硬件打交道，并且通过少量的应用程序编程接口向上层提供一个内核的抽象。这部分包括进程管理、进程间通信、基本内存管理以及中断的响应框架，具体功能包括：

(1) 进程管理。负责实现进程的创建、进程的撤销以及最为核心的进程调度。

(2) 进程间通信。负责提供一种或多种进程间直接或间接通信的方式。

(3) 基本内存管理。负责对物理内存进行分配与回收，以及对进程虚拟地址空间的管理。

(4) 陷入处理。负责整个系统的系统调用、异常与中断响应框架的处理，这是系统其余部分正常工作的基础。

2.2.2 服务器部分

这部分实现的是高层服务的提供者，它们以服务器的形式存在于用户空间中，并且采用客户机/服务器的方式与普通用户进程进行通信。服务器通过消息中指定的操作符与参数调用执行本模块实现的系统服务。采用这种设计方案可以避免一个服务器的行为干扰同一系统中的其他模块，每个服务器仅仅公开必须要让外界知道的接口即访问方式而将具体的实现细节隐藏了起来。

2.3 开发平台

系统开发采用了Linux平台，这是因为一方面Linux环境下有着比较齐全的系统开发工具，比较适合做系统层开发平台；另一方面Bochs虚拟机也有相应的运行在Linux平台上的版本。

3 系统实现

3.1 微内核关键机制的实现

本系统微内核主要实现了五个功能模块，除了2.2.1所述的四个功能模块外，还实现了内核代理服务器。它也是一个进程，与其他普通进程的区别仅在于：它没有用户空间堆栈并且始终运行在内核空间中。主要作用是代表内核与核外各服务器以及普通用户进程通信。

对于内核中其他功能模块，我们都是从便于学生理解和扩展出发，实现操作系统原理所述的基本概念。在进程管理模块中，重点设计和实现进程控制块，进程调度采用了基于优先级的FIFO调度算法。系统设计并实现了两种具有代表性的进程间通信机制：消息传递和共享内存，其中消息传递的接收是阻塞式的，发送是同步的。系统以Bochs模拟的i386处理器的分页存储管理为基础实现一个两层的页式管理模型，向学生展示典型的分级页式存储管理。内核为每个进程分配的逻辑地址空间大小为64MB。系统在内核中分别实现了中断、异常与系统调用。这三类事件的处理过程基本相似：内核在接收到中断信号后，在完成底层处理后立即向相应服务器发送消息，然后系统再切回到用户空间。

3.2 典型服务器的实现

基于客户机/服务器体系结构的微内核操作系统中，各个服务器与微内核被分别编译成独立的二进制文件，服务器与内核之间是松耦合的关系。从微内核的角度来看，它们与普通的用户进程没有任何的区别。

所有服务器的基本流程都是一样的。系统初始化时，每一个服务器都将被启动并执行各自的初始化操作，然后阻塞地等待客户消息的到来。当消息到达后服务器调用适当的子模块执行对应操作，完成后向客户发送一条应答消息然后再次阻塞地等待消息，如此不断地循环下去。

目前我们实现了以下两个服务器：

(1) 文件系统服务器。将接收到的对于逻辑文件的处理转化成对存储于物理介质上的物理文件的处理，为用户进程提供一个有效的文件访问方式。为了方便教学，我们实现了一种相对比较简单但同时又能解释清楚其工作原理的文件系统。在此基础上我们很容易设计相应的习题，要求学生对现有文件系统作功能上与性能上的增强。

(2) 磁盘服务器。将接收到的磁盘读写消息转换成对磁盘控制器发送命令的操作，磁盘服务器通过隐藏底层硬件的具体操作向用户进程提供一种有效的磁盘访问方式。

4 系统配置与使用演示

系统目前共实现了6个核心模块，涉及文件25个，共2900行C代码。

4.1 系统配置

由于系统运行在Bochs虚拟机上，因此首先需要配置系统，包括内核、服务器和应用程序。下列脚本

将微内核各模块连接成可执行二进制代码，再将bootloader与编译以后的微内核可执行代码合并成一个可执行文件os.bin作为启动盘使用：

```
ld -Ttext 0x9000 -e main ${INITOBJ}\${OBJ} -o kernel.img -Map os.map
objcopy -R .note -R .comment -S -O binary kernel.img init/kernel.bin
cp boot/boot.bin os.bin
cat init/kernel.bin >> os.bin
```

在bochs配置文件bochs.txt中加入以下这两行：

```
floppya: 1_44=os.bin, status=inserted
boot: a
```

配置完内核以后，接下来还需要对服务器、用户程序按类似步骤进行处理。所有的服务器将被合并到server这个磁盘映像文件中，其他用户程序则被合并到driverc.img这个磁盘映像文件中。然后在bochs.txt中设置：

```
floppyb: 1_44=server, status=inserted
diskc: file=driverc.img, cyl=20, heads=16, spt=63
```

最后是对配置文件中一些其他属性的设置。

4.2 使用演示

正如前文所强调的，本系统较之现有教学工具更为有效之处在于可以方便地、全面地对系统进行调试。以下是以进程切换这一教学难点为例，用bochsdbg观察地址空间与系统堆栈的切换过程。

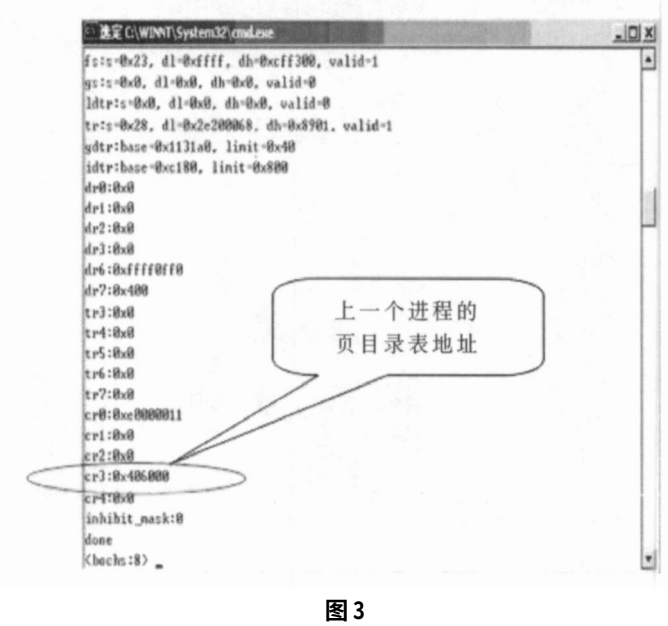

图3

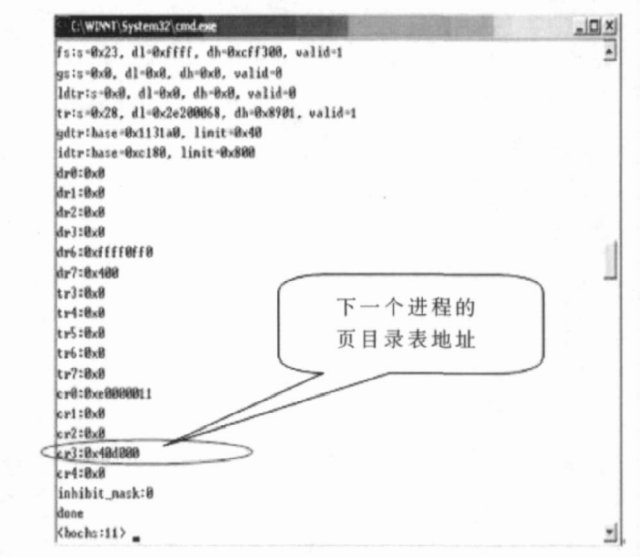

图4

我们在调度器的入口以及进程切换函数的入口分别设置断点，调度器被执行时如图 3 所示，使用 dump_cpu 观察到当前 cr3 寄存器中的值为 0x406000。当执行完

```
_asm_volatile("movl %0, %%cr3"::"r"(global_current->pageTable));
```

以后，再次使用 dump_cpu 命令时可以清楚地看到此时cr3中的值已经被换成了0x40d000，如图4所示。

系统堆栈的切换发生在 SWITCH_TO 中，执行前与执行后用infor观察到的结果分别如图5和图6所示。

5 总结

本课题设计并实现了一个基于Bochs虚拟机的微

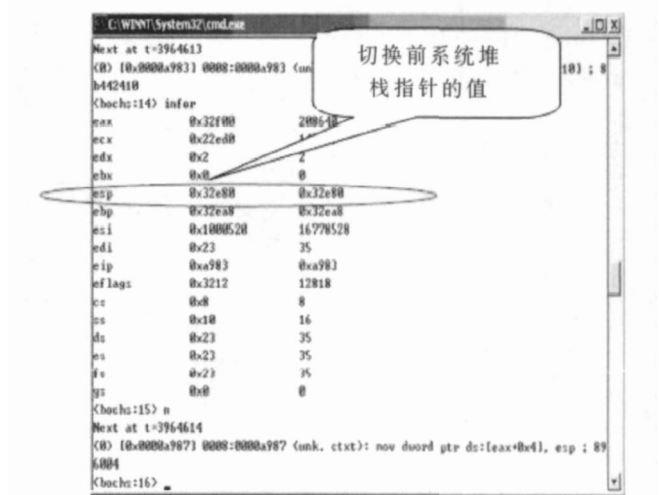

图5

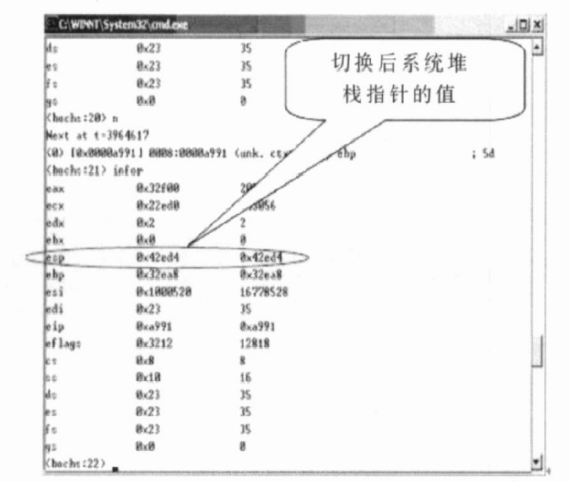

图 6

内核结构的教学用操作系统。该系统在保持简化性的前提下，展现了一个真实操作系统的工作原理与机制，并且保持了一定的可扩展性，从而为学生提供了一个良好的实验平台，在一定程度上弥补了现有操作系统教学工具的缺陷。实现的系统包括两部分：微内核和核外服务器。希望通过这一平台来降低操作系统的学习门槛，帮助学生由浅入深，由表及里，循序渐进地了解和掌握操作系统的原理，从而克服操作系统教学中实践滞后于理论的弊端。

参考文献：

[1] Alfredo Perez Davila. O.S. bridge between academia and reality [EB]. ACM SIGCSE Bulletin, 1995.

[2] Gary Nutt. Linux操作系统内核实习[M]. 机械工业出版社，2004.

[3] Steven Robbins, Kay A. Robbins. Empirical exploration in undergraduate operating systems [EB]. The proceedings of the 13th ACM SIGCSE Bulletin. 1999-3.

[4] 毛德操，胡希明. Linux内核源代码情景分析（上册）[M]. 浙江大学出版社，2001.

[5] 赵炯. Linux内核完全注释[M]. 机械工业出版社，2004.

[6] Andrew S. Tanenbaum Albert S. Woodhull Operating system design and implementation[M]. 清华大学出版社，1997.

[7] Nachos官方网站 http://www.cs.washington.edu/homes/tom/nachos/

[8] 付长冬，孟庆余，潘清. 基于微内核的操作系统综述[J]. 计算机工程与科学，1997(8).

[9] 潘清，张晓清. 操作系统微内核技术研究[J]. 软件学报，1998(8).

[10] Bochs官方网站 http://sourceforge.bochs.com

收稿日期：2006-09-18

作者简介：王红玲（1975-），女，苏州人，苏州大学计算机科学与技术学院，硕士，讲师，主要研究方向：操作系统、中文信息处理。

通信地址：苏州市十梓街 1 号 苏州大学 158 信箱

邮编：215006

E-mail:redleaf@suda.edu.cn

电话：0512-67165762-204