

Design and Implementation of Operating System Assisted Teaching Platform

Jinquang Zhang^{1,2}, Lina Ni^{1,2}, Hua Zhao^{1,2}, Jiancong Fan^{1,2}

¹College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao Shandong

²Shandong Province Key Laboratory of Wisdom Mine Information Technology, Shandong University of Science and Technology, Qingdao Shandong

Email: tjzhangjinquan@126.com, nln2004@163.com

Received: Mar. 6th, 2016; accepted: Mar. 26th, 2016; published: Mar. 29th, 2016

Copyright © 2016 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Operating system is one of the core computer science courses, with a strong experimental feature. It is often difficult to understand and grasp well because the course has many concepts and involves wide aspects of real operating system. A large number of researches have been done in practice teaching of the operating system course at home and abroad. In this paper, we construct a virtual operating system environment to provide a simple, easy to use platform for study, design and research of operating system. The platform has also laid a good foundation for operating system teaching work.

Keywords

Operating System for Teaching, Process Generator, Virtual Platform, Assisted Teaching

操作系统辅助教学平台的设计与实现

张金泉^{1,2}, 倪丽娜^{1,2}, 赵 华^{1,2}, 樊建聪^{1,2}

¹山东科技大学信息科学与工程学院, 山东 青岛

²山东科技大学山东省智慧矿山信息技术省级重点实验室, 山东 青岛

Email: tjzhangjinquan@126.com, nln2004@163.com

收稿日期：2016年3月6日；录用日期：2016年3月26日；发布日期：2016年3月29日

摘要

操作系统是计算机专业的一门核心专业课程，具有较强的实验性。由于概念多、涉及面广等因素，往往难以很好地理解和掌握，国内外在操作系统实践教学方面所做了大量研究。本文通过构造一个虚拟操作系统运行环境，试图为学习、设计和研究操作系统提供一个简单、易用的平台，为操作系统的教学工作打下了良好基础。

关键词

教学用操作系统，进程生成器，虚拟平台，辅助教学

1. 引言

计算机操作系统课的教学一直在计算机学科的教学计划中占据重要地位[1] [2]。在教学中存在主要问题课程内容枯燥难懂，造成学生对课程学习缺乏兴趣，最后以死记硬背应付了事。理论与实际脱节、缺少实例分析，实践性环节薄弱[3]。目前国内外学者开发了许多教学用操作系统，MINIX [4]是荷兰A.S.Tanenbaum教授所发展的一个类Unix操作系统，包括了进程管理、文件系统管理、存储管理、设备管理以及I/O管理等操作系统的所有重要内容，以及系统启动和Shell等实际操作系统不可缺少的部分。但是MINIX作为教学用操作系统有它的不足之处，移植性并不令人满意。GeekOS [5] [6]是一个基于X86架构的PC上运行的微操作系统内核，提供了操作系统与硬件之间的所有必备接口，实现了系统引导，实模式到保护模式的转换，中断调用及异常处理，基于段式的内存管理，FIFO进程调度算法以及内核进程，基本的输入输出，以及一个用于存放用户程序的只读文件系统PFAT。OSP 2系统[7] [8]的核心是一个模拟器，它使用动态演化的、多程序运行的用户进程集来虚拟计算机系统。由一系列的模块组成，每一模块执行一项操作系统服务。可以按照希望的顺序来组织项目，以便与课堂讲义的内容保持同步。Nachos系统[9]是加州大学伯克利分校在操作系统课程中已多次使用的操作系统课程设计平台，采用面向对象的通用虚拟机，确定性调试比较方便，简单而易于扩展。但是Nachos是针对RISC结构MIPS处理器，使用其它开发环境的话，需要使用交叉编译器才能把代码编译成MIPS相应的机器代码。Bochs系统[10]由凯文·劳顿编写的，使用c++编写的开源IA-32(x86)电脑模拟器，它仿真英特尔x86 CPU、常见的I/O设备、和定制的BIOS。MOS系统[11]MOS操作系统是上海交通大学开发的，主要包括作业调度管理和文件系统管理，建立在一个只包含十几条指令的指令集虚拟机基础之上。MOS的不足是过于简单，不能涵盖操作系统的大部分功能。MOS的虚拟机指令集是自定义的，没有现成的编译器，所以读者必须直接编写汇编程序才能在MOS虚拟机上运行。

综上所述，作者认为为了使计算机科学与技术及相关相关专业学生学习理解操作系统的诸多概念，直观地看到进程在执行过程中，资源申请中可能出现的死锁，以及同步、互斥等重要问题的解决过程，有必要研究设计一个对于初学者来说使用更加简单和方便的辅助教学平台。

2. 辅助平台结构设计

作者设计开发的辅助学习系统，包括三个方面的功能：一是模拟操作系统实现过程；二是提交自己的作业到服务器，服务器内核在执行完学生的程序后将结果反馈给学生；三是师生交流论坛，为学生学

习操作系统，解答难题提供进一步的支持。

本文主要描述第一个功能部分，它由三层结构组成，即硬件模拟层、系统运行模拟层和用户应用层。

系统硬件层模拟计算机中包括打印机、外存储器、内存等主要资源的管理。

用户应用层主要包含两个功能：1) 用户程序整合到内核模块。包括进程调度、内存管理、页面置换、设备分配等算法，这些算法用户可以根据操作系统已有算法编写，也可以由用户提出新的算法。然后通过平台提供的接口提交到内核中。2) 系统运行信息查询。主要有系统进程相关信息包括进程号、进程状态、开始时间、已执行时间、已分配内存大小，进程调度顺序，就绪队列，阻塞队列等，系统设备相关信息包括设备请求队列、空设备队列等，以及内存分配情况包括可分配内存、内存中进程分布情况等用户主要关心信息。

系统运行模拟层是本平台的主要部分，主要包括进程生成器、进程调度接口、内存分配接口、设备调度接口、磁盘调度接口等。其中每个接口都实现了一种基本算法，用户可以重载这些接口实现操作系统中的各个算法，当然也可以自己编写新的算法，或直接使用本平台提供的算法。

3. 操作系统运行过程模拟

3.1. 进程生成器

在教学过程中，进程管理是学生难以理解和掌握的难点问题。进程状态之间的转换，以及同步问题难以直观看到，因此，希望通过开发一个进程模拟器将进程在生命周期中，各种可能的资源申请过程、出现的各个状态以及发生的事件直观地描述出来。进程生成器生成虚拟操作系统运行过程中所需要的各个进程，它包含的数据结构为：

1) 进程号，用来区分不同的进程；

2) 进程状态，包括就绪、阻塞、运行、挂起等基本状态；

3) 初始优先级，每个进程分配一个运行的优选顺序；

4) 优先级类型，采用动态优先级；

5) 内存大小，表示系统需要的初始内存，在使用过程中可以动态申请内存，而且本平台中将进程使用内存分为连续和离散两种类型；

6) 已装入部分，其结构为：(装入时间 1，访问地址 1)、(装入时间 2、访问地址 2)、...、(装入时间 k，访问地址 k)；初值均为-1。；

7) 已执行时间，进程使用 CPU 的时间，用来判断进程是否执行结束；

8) 已等待时间，处于就绪状态的进程在就绪队列中的时间，用来计算进程的动态优先级；

9) 用户优先级，分配给用户的初始优先级，为用户进程的调度提供依据；

10) 进程分段数。由于计算所做的工程可以描述为计算、输入输出、计算、输入输出，...等，因此将进程分成若干段，即：(计算、计算时间 1)、(I/O、I/O 时间 1)、(计算、计算时间 2)、(I/O、I/O 时间 2)、...、(计算、计算时间 n)；其中，计算时间 $i(i=k)$ ，结构为：时间、访问的地址(地址 1、地址 2、...、地址 k)；I/O 时间 i 的结构为：时间、输入\输出。

11) 需要资源，其结构为：(资源名 1，使用起始时刻 1，使用时间 1)、(资源名 2，使用起始时刻 2，使用时间 2)、...、(资源名 k，使用起始时刻 k，使用时间 k)

在初始化时，预先生成大量的进程存储在备份结构中，在运行时，随机生成新的进程，用来模拟新作业进入系统执行。

3.2. 系统运行过程

当用户运行辅助平台时首先进行初始化工作，然后执行所设计的算法，模拟操作系统的运行过程。初始化过程包括：

1) 构建空的“就绪”进程队列、“阻塞”进程队列、“挂起”进程队列、“终止”进程队列、I/O 请求队列、正在进行 I/O 的进程队列、I/O 空设备队列；初始一个时钟，便于进程的调度。

2) 进程号初值(PID)、最大进程号(MAXID)。

3) 选择进程调度算法，初始默认使用先来先服务算法，在用户实现接口后就可以选择不用的算法，主要有①先来先服务、②短作业优先、③优先权调度、④时间片轮转、⑤多级队列和⑥多级反馈队列等算法。正如前面描述的那样，用户还可以增加自己新的算法。

4) 选择内存分配算法。在平台中默认用户空间大小为 1 GB，在生成进程通过一个随机函数生成 1~8 之间的一个随机数 a ，分配给进程的初始内存为 aMB 。然后分成连续内存分配和分页式内存分配方式。在连续内存分配中实现了①首次适应、②循环首次适应、③最佳适应和④最差适应；在采用分页内存方式中实现了①先进先出、②最佳页置换、③最近最久未使用、④二次机会、⑤改进的二次机会、⑥最少使用和⑦最多使用等算法。

5) 选择设备分配算法，系统中的设备采用先来先服务和优先权服务等两种方式分配。

6) 创建进程，将预先准备的进程装入内存的过程，包括分配进程号和调用“内存分配算法”接口分配内存。如果允许，则将进程挂入就绪进程队列。否则挂入“挂起”队列。

初始化过程结束后，执行以下算法模拟操作系统的运行过程：

Step 1: 调用“进程调度算法”从“就绪”队列中选择一个可执行进程，如果存在，则将选中的进程状态改为“执行”。返回值(进程号、执行时间、访问内存地址)

Step 2: 对每一个执行时间单位：

Step 2.1: 检查相应内存地址是否可访问，如果不能访问，则

- 1) 将进程的状态改为“阻塞”；
- 2) 调用“内存置换算法”；
- 3) 调用“I/O 设备请求分配算法”，分配设备；
- 4) 调度“中断处理算法”，模拟中断的处理过程；
- 5) 时钟加 1
- 6) 转 Step 1;

Step 2.2: 如果能访问，则

- 1) “延长一段时间”模拟程序的执行；
- 2) 当前“计算时间”减 1；
- 3) 已执行时间加 1；
- 4) 删除所访问的地址；
- 5) 时钟加 1
- 6) 调度“中断处理”算法，模拟中断的处理过程；

Step 3: 检查当前进程：

如果当前剩余“计算时间”不为 0，则将状态改为“就绪”；

否则，则检查“下一个时段”，如果为空，则将状态改为“终止”，并挂入“终止”队列；

否则，将状态改为“阻塞”，调用“I/O 设备请求分配算法”，分配设备；

Step 4: 返回到 Step 1。

其中进程调度算法、内存置换算法、I/O 设备请求分配算法, 由用户编写, 然后提交到平台, 再整合到内核中。在本次实现的平台中, 进程调度算法每次运行提交一种调度算法, 如果要使用其它调度算法则需要重新提交(自动覆盖原来的算法), 内存置换算法和 I/O 设备请求分配算法的处理方式与进程调度一样。“中断处理算法”模拟中断的处理过程。首先通过一个随机函数产生一个随机数决定是否要立即响应中断。如果不立即响应中断则继续调度程序执行。否则, 处理中断, 执行过程如下:

- 1) 检查是否有 I/O 完成的进程, 如果有就其挂在“就绪”队列上;
- 2) 检查是否有“终止”进程, 如果有, 则回收资源(主要是内存), 并检查“挂起”队列中是否有等待创建的进程, 如果可以则创建。
- 3) 通过生成随机数的方式, 决定是否要生成一个新进程(模拟用户生成新进程)。

4. 结语

本文首先讨论了国内外学者在如何提高操作系统的教学效果, 尤其是在实践教学环节上的探索。笔者针对在教学过程中, 各个层次的学习情况, 设计一个辅助教学平台, 将抽象的进程申请资源, 内存分配等主要问题, 直观的展示出来。使该教学辅助平台成为学生学习操作系统的一个重要帮手, 为实现操作系统的教学目标提高保障。下一步的研究重点是, 探讨在伪代码上实现同步问题的判断。

基金项目

本文得到山东科技大学教育教学群星计划项目(qx102043)的资助。

参考文献 (References)

- [1] 蔡启先. 《CC2001 教程》研究与思考[J]. 比较教育研究, 2003(2): 65-70.
- [2] 袁开榜. 计算机学科教学计划(1993) [J]. 计算机世界报, 1994(39): 1-8.
- [3] 包向辉, 尚晓丽. 计算机操作系统教学方法的探讨与改进[J]. 硅谷, 2008(22): 145-145.
- [4] Minix 维基百科[EB/OL]. <http://zh.wikipedia.org/wiki/Minix>
- [5] 黄廷辉, 王宇英. 计算机操作系统实践教程[M]. 北京: 清华大学出版社, 2007: 1-120.
- [6] Hovemeyer, D. GeekOS: An Instructional Operating System for Real Hardware. <http://geekos.sourceforge.net/docs/>
- [7] William Stallings. 操作系统—精髓与设计原理(第五版) [M]. 北京: 电子工业出版社, 2006: 20-112.
- [8] Michael Kifer 著, 王俊华, 等, 译. 操作系统设计与实现[M]. 北京: 清华大学出版社, 2010: 1-230.
- [9] <http://www.cs.washington.edu/homes/tom/nachos/>
- [10] <http://baike.baidu.com/view/31379.htm>
- [11] 李红卫, 郭庆军, 殷常鸿. 操作系统原理与实践教程[M]. 北京: 科学出版社, 2008: 1-160.