**山东大学 计算机科学与技术 学院**

**《 操作系统课程设计 》教学大纲**

编写人：杨兴强、韩芳溪 审定人：贾智平

编制时间：2017.6.7 审定时间：2017.8.20

**一、课程基本信息：**

|  |  |  |  |
| --- | --- | --- | --- |
| 课程名称 | 操作系统课程设计 | | |
| 英文名称 | Operating System Curriculum Design | | |
| 课程编码 | 01330211 | | |
| 开课单位 | 计算机科学与技术学院 | | |
| 课程类别 | □通识教育必修课程 □通识教育核心课程  □通识教育选修课程 □学科基础平台课程  □专业基础课程 √专业必修课程 □专业选修课程 | | |
| 课程性质 | √必修 □选修 | | |
| 学分 | 2 | 学时 | 16+32 |
| 适用专业 | 计算机科学与技术 | | |
| 先修课程 | 操作系统、高级语言程序设计 | | |
| 课程网站 | http://course.sdu.edu.cn/G2S/Template/View.aspx?action=view&courseType=0&courseId=1487 | | |

**二、课程描述**

操作系统课程设计是计算机科学与技术专业的实践必修课，该课程以一个操作系统Nachos的设计与实现过程为例，阐述操作系统软件中具有普遍意义的方法与结构，使学生能够深入理解操作系统的工作原理与实现过程，培养学生系统软件的开发能力。课程要求学生在分析Nachos系统源代码的基础上，按指定要求，设计解决方案，编程扩展、完善系统的功能。

课程由课堂教学和实践两部分构成：课堂教学介绍Nachos操作系统的整体架构，引领学生通过分析nachos系统代码，理解Nachos的总体架构及功能实现，设计Nachos系统的修改方案；实验部分根据设计的方案由学生自己动手编程实现课程目标。

课程的特色是理论联系实际。基于操作系统的理论与方法对Nachos相关模块进行编程实践，在培养学生系统设计编程能力的同时，加深学生对操作系统运行机制的深入理解，同时也培养了学生按照软件工程的观点，基于现代化的工具软件，分析、设计与开发大型系统软件的能力及团队合作精神。

The operating system curriculum design is a compulsory course for computer science and technology majors. This course takes the design and implementation process of an operating system (Nachos) as an example to illustrate the methods and structures used in the operating system software, so as to enable students to deeply understand the operating system working principle and implementation process and method, as well as cultivate the ability of student to develop the system software. The course requires students to work out the design for extending and improving Nachos functions, and then program it in C++.

The course consists of two parts: lectures and practices. In lectures, students will be lead to analyze the nachos system code, in order to understand the overall architecture and function realization of Nachos. During the experimental part, students will design and program the modification scheme to improve the Nachos system so as to meet objective of this course.

This course combines theory with practice. It will not only deepen students’ understanding of how the operating system works but also cultivate the students’ capability to analyze, design and develop large scale system software, as well as their teamwork spirit, according to the idea of software engineering and using modern software tools.

**三、课程教学目标和教学要求**

【教学目标】

培养学生分析大型系统程序代码的能力，提高学生对计算机领域复杂工程问题程序的设计、开发、调试及保证系统健壮性的能力；培养学生对大型系统软件的分析、设计与开发能力，针对课程的具体要求设计相应的解决方案，并在实践中不断对方案进行优化；在系统的开发过程中，培养学生使用现代编程工具与调试工具的能力，并能充分利用其优势克服其局限性；结合大型系统的开发与调试过程，培养学生将工程管理理论和方法应用在实践中解决复杂工程问题的能力

【教学要求】

|  |  |  |
| --- | --- | --- |
| 毕业要求 | 指标点 | 课程教学目标 |
| 1.工程问题 | 1-4能够将数学、自然科学、工程基础和专业知识等综合应用于解决计算机领域复杂工程问题。 | 目标1：培养学生分析大型系统程序代码的能力，提高学生对计算机领域复杂工程问题程序的设计、开发、调试及保证系统健壮性的能力。 |
| 3.设计/开发解决方案 | 3-4 在充分理解计算机软硬件及系统的基础上，能够设计针对计算机领域复杂工程问题的解决方案，设计或开发满足特定需求和约束条件的软硬件系统、模块或算法流程，并能够进行模块和系统级优化； | 目标2：培养学生对大型系统软件的分析、设计与开发能力，针对课程的具体要求设计相应的解决方案，并在实践中不断对方案进行优化。 |
| 5.使用现代工具 | 5-3 善于利用所使用的技术、资源和工具的优势，并能分析和把握其局限性，以更好的用于复杂工程求解。 | 目标3：在系统的开发过程中，培养学生使用现代编程工具与调试工具的能力，并能充分利用其优势克服其局限性。 |
| 11.项目管理 | 11-3 能够在多学科环境中应用工程管理原理与经济决策方法，能将工程管理理论和方法应用在实践中解决复杂工程问题，具备初步的计算机工程项目管理经验与能力。 | 目标4：结合大型系统的开发与调试过程，培养学生将工程管理理论和方法应用在实践中解决复杂工程问题的能力。 |

**四、课程教学内容及学时分配**

第一章 Nachos虚拟机（4学时授课+2学时上机）

本章目标：

1. 描述Nachos的基本组成
2. 说明Nachos虚拟机的模拟方法
3. 熟悉Nachos的安装过程及调试方法

第一节 Nachos的硬件设备

【教学目标】通过了解Nachos机器的各种硬件设备，形成Nachos计算机的整体轮廓。

【教学要求】有选择地介绍Nachos硬件。

【教学内容】计时器，中断，磁盘，I/O设备，指令系统。

【教学和学习建议】讲授、讨论。

【难点重点】整体把握Nachos的硬件架构。

第二节 Nachos虚拟机

【教学目标】掌握各种硬件的软件模拟方法

【教学要求】针对每种Nachos硬件，说明相应的软件模拟方法，分析虚拟机器与真实机器的异同，以及对后续系统实现的影响

【教学内容】结合machine目录中的相关代码，介绍Nachos对时钟、中断、分时、I/O设备、磁盘的实现。

【教学和学习建议】讲授、讨论。

【难点重点】硬件设备的实现过程，分析是否足够真实地模拟了硬件环境

实验一 Nachos安装与调试（2学时）

【教学目标】熟悉Nachos的安装及调试过程，了解Nachos对计算机硬件的模拟方法。

【教学要求】熟悉Nachos的安装过程，了解相关的makefile文件，熟悉程序调试环境gdb，熟悉交叉编译的环境设置，了解Nachos实现的中断控制器、定时器等设备的工作过程。

【教学内容】安装Nachos及交叉编译器，熟悉Nachos中makefile以及gdb使用，利用gdb跟踪几个模拟设备的工作过程。

【教学和学习建议】上机、讨论。

【难点重点】交叉编译器及Nachos的安装过程，基于gdb的调试过程；交叉编译器及中断控制器工作过程的理解

1. 线程管理（4学时授课+8学时上机）

本章的目标：

1. 描述Nachos线程的概念及实现
2. 线程的有关操作
3. Nachos线程上下文切换
4. Nachos信号量及线程之间的同步

第一节 线程包

【教学目标】Nachos线程的管理。

【教学要求】了解Nachos线程管理的各项功能。

【教学内容】线程创建、线程切换、线程睡眠、线程终止、线程调度、执行模式，上下文切换。

【教学和学习建议】授课、讨论。

【难点重点】从整体上把握线程包的功能及其之间的关联。

第二节 Nachos线程创建

【教学目标】了解Nachos线程创建的过程。

【教学要求】分析源代码，理解Nachos线程的概念及其创建过程。

【教学内容】结合main.cc、thread.cc、system.cc以及threadtest.cc等模块，分析系统初始化及线程的创建方法。

【教学和学习建议】授课、讨论。

【难点重点】Nachos内核初始化以及线程的创建过程。

第三节 Nachos线程调度及上下文切换

【教学目标】理解Nachos线程调度及线程切换程序的运行过程。

【教学要求】分析Nachos中线程调度算法、实现过程，以及上下文切换的具体实现。

【教学内容】结合scheduler.cc、thread.cc及switch-linux.s，分析其设计实现方法。

【难点重点】线程切换的工作过程及实现方法。

第四节 Nachos线程同步

【教学目标】理解Nachos中信号量工作原理及线程同步机制。

【教学要求】分析Nachos中信号量及管程的实现方法，并利用其实现进程之间的同步。

【教学内容】结合synch.cc、synchlist.cc、prodcons++.cc及ring.cc等模块，分析Nachos中信号量构造、创建及使用方法，并利用信号量实现生产者-消费者模型。

【教学和学习建议】授课、讨论。

【难点重点】信号量及管程的工作机制及实现方法

实验二 Nachos中线程的上下文切换（2学时）

【教学目标】了解Nachos中线程上下文切换的工作机理

【教学要求】通过gdb调试器跟踪Nachos中内核线程的调度与上下文切换过程

【教学内容】将nachos加载到gdb中，在gdb中适当位置设置断点，利用gdb提供的命令跟踪Nachos中线程的切换过程，对跟踪结果进行分析；

【教学和学习建议】上机、讨论。

【难点重点】线程调度方法及上下文的切换过程

实验三 信号量同步（6学时）

【教学目标】了解Nachos中利用信号量实现线程同步的工作机理

【教学要求】利用信号量机制完成生产者-消费者问题，并跟踪其运行过程

【教学内容】分析main.cc、prodcons++.cc，ring.cc、ring.h与synch.cc文件，利用Nachos实现的同步接口，编程实现生产者-消费者问题，并跟踪其运行过程。

【教学和学习建议】上机、讨论。

【难点重点】线程创建方法及信号量的使用方法，利用信号量解决线程的同步问题

1. 文件系统管理（4学时授课+8学时实验）

本章目标：

1. 了解文件系统的工作原理
2. 了解文件系统管理的实现、相关数据结构的组织及其关联
3. 完善Nachos文件系统，使其能够支持文件的扩展功能

第一节 文件管理包

【教学目标】Nachos文件系统的管理。

【教学要求】了解Nachos文件系统管理包的各项功能。

【教学内容】文件创建、文件系统结构等。

【教学和学习建议】授课、讨论。

【难点重点】从整体上把握文件管理包的功能以及类之间的关联。

第二节 文件系统的功能与实现

【教学目标】了解Nachos文件系统的管理功能，相关的数据结构。

【教学要求】了解Nachos文件系统的现有功能，实现方法

【教学内容】分析fstest.cc、synchdisk.cc、openfile.cc、filesys.cc、directory.cc、filehdr.cc、fstest.cc与Bitmap.cc等文件，了解Nachos磁盘的创建与初始化方法，文件系统的设计、实现方法及相关的数据结构（如文件目录表、FCB、inode、空闲块位示图等）

【教学和学习建议】授课、讨论。

【难点重点】文件系统相关数据结构及其之间的数据关联

实验四 Nachos文件的组织与结构（2学时）

【教学目标】通过跟踪调试nachos文件系统，掌握Nachos文件系统的工作机理

【教学要求】了解Nachos中文件系统所实现的功能、实现方法及文件系统的相关数据结构

【教学内容】在分析../code/filesys目录下有关文件管理几个模块的基础上，结合hexdump -C filename、nachos –D等命令，了解Nchos文件系统的实现细节及相关的数据结构的组成及其关联（如文件目录表、FCB、inode、磁盘空闲块Bitmap等）。

【教学和学习建议】上机、讨论。

【难点重点】文件系统在磁盘上的布局，文件头（inode，FCB）、目录项与目录表、空闲块位示图等相关的数据结构及其关联；

实验五 可扩展文件的设计与实现（6学时）

【教学目标】对nachos中基本文件系统功能进行完善，设计实现文件的动态扩展功能

【教学要求】修改文件系统的相关代码，以支持根据文件的大小动态分配磁盘块数。

【教学内容】在实验四的基础上， 阅读修改OpenFile::WriteAt()、FileHeader::Allocate()与fstest.cc中Copy()、Append()、NAppend()等函数，使文件系统具有根据文件大小动态扩展的功能。

【教学和学习建议】上机、讨论。

【难点重点】将文件系统的相关数据结构进行整合，实现文件的可扩展功能。

第四章 应用程序与内存管理（4学时授课+14学时上机）

本章目标：

1. 理解用户程序和操作系统的关系
2. 理解应用程序的加载过程及执行过程。
3. 完善Nachos，使其支持多进程管理。
4. 实现相应的系统调用。

第一节 内存管理包

【教学目标】Nachos内存管理

【教学要求】了解Nachos内存管理包的各项功能。

【教学内容】Nachos应用程序及内存管理。

【教学和学习建议】授课、讨论。

【难点重点】从整体上把握内存管理包的全部功能；

第二节 Nachos应用程序

【教学目标】创建Nachos应用程序的方法及应用程序调用内核提供服务的方法。

【教学要求】如何基于交叉编译程序建立Nachos能够运行的MIPS应用程序。

【教学内容】交叉编译环境的建立，应用程序结构，应用程序转换器coff2noff.c，Nachos系统调用的使用与参数传递；

【教学和学习建议】授课、讨论。

【难点重点】编写Nachos应用程序的语法，交叉编译生成应用程序可执行文件的方法及系统调用的使用

第三节 Nachos地址空间的管理

【教学目标】理解操作系统加载应用程序的过程，并为其分配内存空间的方法。

【教学要求】结合代码分析Nachos对存储器的管理方法。

【教学内容】分析addrspace.cc、progtest.cc、bitmap.cc等文件，理解Nachos如何将一个用户程序加载到内存，为其分配相应的内存空间，创建页表与建立虚页与实页的映射关系，以及将一个用户线程映射到核心线程与将应用进程的页表映射到系统页表的理念与方法。

【教学和学习建议】授课、讨论。

【难点重点】应用程序的加载过程、内存空间的分配及虚实地址的转换方法。

第三节 Nachos应用程序的执行过程

【教学目标】理解Nachos启动执行应用程序的过程。

【教学要求】结合代码说明Nachos启动应用程序及解析执行应用程序指令的方法。

【教学内容】分析machine.cc、progtest.cc、schduler.cc等文件，理解Nachos将一个用户程序加载到内存并为其分配相应的内存空间后，如何启动该应用程序并解析执行其指令的过程，以及应用进程的上下文切换过程。

【教学和学习建议】授课、讨论。

【难点重点】Nachos的CPU对应用程序指令的解析执行过程

第四节 Nachos多进程并发技术的设计与实现

【教学目标】理解操作系统对多道程序设计技术的设计与实现方法。

【教学要求】修改Nachos代码，提供相应的系统调用接口，使Nachos具备多进程并发执行的功能。

【教学内容】分析修改addrspace.cc、exception.cc等相关程序，实现系统调用Exec()及Exit()，使Nachos能够为每道程序分配独立的内存空间，为新建进程分配进程号，并基于页表实现虚实地址的转换，在进程正常退出时能够回收其释放的内存空间等资源。

【教学和学习建议】授课、讨论。

【难点重点】内存空间的分配及内存保护措施。

实验六 用户程序与系统调用（2课时）

【教学目标】理解用户程序加载及执行过程，系统调用的工作原理。

【教学要求】理解如何创建用户进程，用户进程是如何利用系统调用与Nachos系统内核进行交互，Nachos系统调用是如何实现的。

【教学内容】 创建Nachos应用程序的方法；在分析addrspace.cc、progtest.cc等文件的基础上，理解Nachos何加载并执行应用程序的方法；理解内核与应用程序中系统调用的交互过程。

【教学和学习建议】上机、讨论。

【难点重点】应用程序的加载过程，应用进程的创建过程，为应用程序分配内存空间的方法，解析执行用户程序机器指令的过程。

实验七 地址空间扩展（6课时）

【教学目标】理解操作系统多道程序设计技术的工作原理、内存分配方法及虚实地址的转换方法。

【教学要求】理解页表生成机制，修改nachos空间分配代码，实现对Nachos对多进程空间分配的支持。

【教学内容】熟悉页式内存管理方法，分析修改addrspace.cc，为每道程序可分配独立的内存空间；

【教学和学习建议】上机、讨论。

【难点重点】应用进程的创建过程，虚实地址的转换及地址分配方法。

实验八 用户程序与系统调用（6课时）

【教学目标】理解多道程序设计技术的实现，系统调用的实现方法。

【教学要求】扩展nachos系统调用相关代码，实现exec()和exit()系统调用，使Nachos支持多线程并发执行。

【教学内容】在实验七的基础上，实现Exec()及Exit()系统调用，使Nachos可同时加载多个应用程序并正常退出，使Nachos可支持多道程序设计技术，并对实验七的设计方案的合理性进行验证。

【教学和学习建议】上机、讨论。

【难点重点】多道程序设计技术的实现方法。

**五、考核及成绩评定方式**

【考核方式】：平时考核、设计报告、现场演示及过程讲解

【成绩评定】：平时考核（20%）、设计报告（30%）、现场演示及过程讲解（50%）

【考试大纲】

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **考试目的** | | 1.考察学生掌握和应用知识的能力**√**2.评价教师教学质量 3.分级教学 4.招生 | | | | | | | | |
| **考试对象** | | 年级： 专业：计算机科学与技术 | | | | | 卷面总分 | | | 100 |
| **考试方式** | | 1.闭卷2.开卷3.上机**√** 4.综述 5.论文 6.设计**√** 7.其它（ ） | | | | | | | | |
| **考试方法** | | 1.笔试 2.口试**√** 3.实际操作**√** 4.其它（ ） | | | | | | | | |
| **试卷来源** | | 1.试题库 2.试卷库 3.校内统一命题 4.校外教师命题 5.任课教师命题**√** | | | | | | | | |
| **试题难易度** | | 1.较容易（ 30 ）% 2. 中等难度（ 50 ）% 3.较大难度（ 20 ）% | | | | | | | | |
| **编**  **题**  **计**  **划** | 题 类  量 型  内 %  容 | | | 记忆 | 理解  分析 | 综合  应用 | | 提高  扩展 | 合计 | |
| 第一章 | | | 5 | 0 | 0 | | 0 | 5 | |
| 第二章 | | |  | 5 | 5 | | 0 | 10 | |
| 第三章 | | |  | 15 | 15 | | 10 | 40 | |
| 第四章 | | |  | 10 | 20 | | 15 | 45 | |
| 合 计 | | | 5 | 30 | 40 | | 25 | 100 | |
| **阅卷方法：** | | | 1.微机阅卷 2.流水阅卷 3.任课教师阅卷**√** | | | | | | | |
| **记分方式：** | | | 1.百分制 2.五级制**√** 3.二级制 | | | | | | | |
| 备注：平时考核（20%）、设计报告（30%）、现场演示及过程讲解（50%）  总成绩 = 平时考核成绩+设计报告评价+现场演示及过程讲解 | | | | | | | | | | |

【教学目标达成与评价】

本课程对课程教学目标的支撑依据，以及对教学目标达成的考核评价方式如下表所示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 课程教学目标 | 支撑依据 | 考查方式与考查点 | 占比 | |
| 1 | **目标1**：培养学生分析大型系统程序代码的能力，提高学生对计算机领域复杂工程问题程序的设计、开发、调试及保证系统健壮性的能力。 | * 依据所学操作系统的理论与方法，通过引领学生分析Nachos操作系统的实现代码，让学生理解操作系统内核启动过程、线程管理、CPU调度、内存管理、文件管理及设备管理的设计与实现方法，培养学生分析大型系统程序代码的能力； * 通过分析系统实现代码，学习并领会系统软件的编程风格、程序模块的组织结构、提高程序可读性及健壮性的措施； * 运用Nachos线程管理及信号量机制，实现生产者消费者模型，要求多个生产者线程与多个消费者线程能够正确并发执行，输出期望的结果；修改完善Nachos的文件系统，设计对文件进行数据追加的解决方案并编程实现，提高学生对计算机领域复杂工程问题程序的设计、开发、调试能力； * 精心设计测试用例对实现的系统进行测试，培养学生保证系统健壮性的能力。 | 系统分析报告、设计报告、编程代码、测试报告及调试过程文档 | 10% | 30% |
| 实时查看学生的设计方案、实现方法、测试过程与调试过程，对发现的问题及时提出修改意见 | 10% |
| 现场演示、过程讲解及问题回答 | 10% |
| 2 | **目标2**：培养学生对大型系统软件的分析、设计与开发能力，针对课程的具体要求设计相应的解决方案，并在实践中不断对方案进行优化。 | * 综合运用Nachos系统的进程管理、线程管理、文件系统、内存管理及I/O管理功能，设计与实现系统调用Exec()与Exit()，使Nachos具备多进程并发执行的功能； * 能力比较强的学生可以进一步设计实现文件的创建、打开、读写及关闭等几个系统调用； * 高水平的学生可再进一步设计实现在用户进程中创建多个用户线程及父进程等待子进程结束的功能； * 精心设计测试用例对实现的系统进行测试 | 系统分析报告、设计报告、编程代码、测试报告及调试过程文档 | 10% | 40% |
| 实时查看学生的设计方案、实现方法与调试过程，对发现的问题及时提出修改意见 | 10% |
| 现场演示、过程讲解及问题回答 | 20% |
| 3 | **目标3**：在系统的开发过程中，培养学生使用现代编程工具与调试工具的能力，并能充分利用其优势克服其局限性。 | * 利用make工具组织大型的工程文件 * 利用gcc与g++与其它的IDE对Nachos系统进行编译 * 利用gdb等对系统进行跟踪调试 | 编程代码、调试过程及调试报告 | 5% | 15% |
| 过程跟踪，实时查看学生的设计方案、实现方法与调试过程，对发现的问题及时提出修改意见 | 5% |
| 现场演示、过程讲解及问题回答 | 5% |
| 4 | **目标4**：结合大型系统的开发与调试过程，培养学生将工程管理理论和方法应用在实践中解决复杂工程问题的能力。 | * 对于Nachos这样一个复杂且庞大的系统，要求学生采用工程管理的理念进行管理 * 按照软件工程的观点，要求学生按步骤提交相应的系统分析、设计、实现、调试、测试及维护报告 | 系统分析、设计、实现、调试、测试及维护文档 | 10% | 15% |
| 过程跟踪，对发现的问题及时提出修改意见 | 5% |

**六、教材及参考资料**

【教材】

作者：Abraham Silberschatz

书名：Operating System Concepts, ninth edition

出版社：Wiley

出版年：2012

【参考书】

现代操作系统

Modern Operating System

Authors: ANDREW S. TANENBAUM

Linux内核设计与实现

Robert Love, 译 陈莉君

伯克利分校nachos课程设计网址

http://www-inst.eecs.berkeley.edu/~cs162/sp14/

C++:http://homes.cs.washington.edu/~tom/nachos/

C++:http://people.cs.uchicago.edu/~odonnell/OData/Courses/CS230/NACHOS/reading-code.html

C++:http://www.cs.duke.edu/~narten/110/nachos/main/main.html

Java: http://http.cs.berkeley.edu/~tea/nachos/nachos.ps

Nachos源代码

http://course.sdu.edu.cn/G2S/Template/View.aspx?courseId=2296&topMenuId=177898&action=view&type=&name=&menuType=2&curfolid=26745

课程设计实验指导手册（自己编写）