

操作系统课程教学大纲

一、课程基本信息

课程代码：BT0300009X0

课程名称（中文）：操作系统

课程名称（英文）：Operating System

课程类别：专业基础必修课

总学时：64 学时（理论 48 学时，实验 16 学时）

学分：4

适用专业：软件工程

先修课程：数据结构与算法，计算机组成原理

二、课程简介

本课程是软件工程专业的专业基础必修课。操作系统是计算机软件系统的核心，它管理和控制计算机系统的硬件与软件资源，是最重要的系统软件。本课程围绕计算机系统的四大类资源管理展开，包括处理器管理、内存管理、设备管理、文件管理的基本原理。通过本课程的学习，学生能理解操作系统在计算机系统的地位和作用，了解操作系统的发展历史、最新发展，能掌握操作系统进程管理、内存管理、设备管理、文件管理的基本原理，能理解进程同步与并发机制，能对操作系统中的经典算法进行分析。在实际软件系统设计时，能站在系统软件的高度思考问题，能充分考虑各种制约因素选用和设计相应的处理模型、算法，具备利用操作系统提供的接口开发具有特定功能应用软件系统的能力。

三、课程目标

课程目标 1. 深入理解操作系统基本原理和理论，能运用操作系统专业知识分析复杂工程问题的解决方案，能综合考虑多种制约因素进行折中获得方案或性能优化。（支撑毕业要求指标点 1.4）

课程目标 2. 能理解计算机系统资源管理的典型算法和调度模型，能对不同算法和模型进行分析和评价，具备根据实际问题需要选用或设计适当的算法和调度模型的能力。（支撑毕业要求指标点 2.2）

课程目标 3. 能配置虚拟机环境并在虚拟机安装 Linux 系统，能理解虚拟机环境的局限性。能使用操作系统接口实现进程创建、进程调度、进程并发、进程通信、内存管理等模型或算法模拟，能对实验结果进行分析。（支撑毕业要求指标点 5.3）

表 1 课程目标与毕业要求指标点对应关系

课程目标	支撑的毕业要求指标点
课程目标 1	1.4 能应用相关知识比较、分析应用型复杂软件系统的解决方案，提出优化和改进建议。

课程目标 2	2.2 能分析应用型复杂软件系统的影响因素，选用或建立适当的模型，通过模型评价获得有效结论。
课程目标 3	5.3 能对模拟、仿真和预测获得的数据进行分析，能理解工具、技术等局限性。

四、基本要求

操作系统是一种系统软件，操作系统所涉及的问题本身就是复杂软件工程问题。为解决处理器管理、内存管理、设备管理等问题都需通过深入分析，建立恰当的抽象模型，并基于模型进行分析和优化，许多问题都需要综合考虑多种制约因素，进行折中才能获得全局优化和较好的系统性能。本课程要求学生能深入理解操作系统运行机制，理解操作系统基本原理、资源管理的基本策略和方法，课程内容应围绕处理器、内存、I/O外设、文件等计算机四大类资源的管理策略展开，为更好的理解典型设计模型和调度算法，应通过实验环节让学生对算法和模型进行模拟，以加深学生对操作系统原理的理解，提高学生分析和解决复杂软件系统问题的能力。

教师可通过课堂提问、课后作业、辅导答疑、课堂测试等环节了解学生学习效果，并根据实际情况调整授课计划和教学活动，保证课程结束时课程目标的达成。

五、理论教学内容

（一）操作系统概论

1. 主要内容

- （1）操作系统的概念、地位、作用，操作系统主要功能与特性；
- （2）操作系统的发展历史和分类，多道程序设计与管理系统的形成；
- （3）操作系统的基本接口和系统调用；
- （4）操作系统的结构和运行模型；
- （5）流行操作系统介绍。

2. 支撑的课程目标

通过本单元的学习，学生能了解操作系统在计算机系统中的地位和作用、操作系统的发展历史、多道程序设计原理、有代表性的操作系统的结构和模型，理解操作系统为用户提供的接口，能从系统软件的高度讨论问题，深化模块划分的系统设计思想，可部分支撑**课程目标 1**。

（二）处理器管理

1. 主要内容

- （1）处理器状态和中断技术；
- （2）进程概念、状态及转换；
- （3）多线程环境中的进程与线程，线程的实现；
- （4）处理器调度层次、选择调度算法原则，作业调度和低级调度算法；

(5) Linux 系统进程管理、进程调度算法。(自学: 2 学时)

2. 重点、难点

重点: 进程状态及转换; 选择调度算法的原则; 常用的作业或进程调度算法;

难点: 常用调度算法的验证, 影响调度算法效率的因素分析。

3. 支撑的课程目标

通过进程与线程、线程并发的学习, 学生能掌握进程、线程基础知识, 理解多线程环境中的进程与线程的实现, 学会在软件系统中使用多线程并发技术提高效率, 可部分支撑**课程目标 1**,

通过调度模型、调度算法的学习, 学生能理解常用作业/进程调度算法, 能分析调度算法的衡量因素, 能在实验中进行验证和模拟。具备在具体应用软件系统设计时根据要求选用和设计恰当算法的能力, 可部分支撑**课程目标 2**。

(三) 进程同步、通信与死锁

1. 主要内容

- (1) 并发程序设计、进程的竞争与协作;
- (2) 临界区管理原则及实现方法;
- (3) 同步机制、信号量和 PV 操作;
- (4) 经典的进程同步问题;
- (5) 管程的实现和利用管程实现进程同步;
- (6) 进程通信机制, 包括信号、管道、共享内存和消息传递通信机制;
- (7) 死锁及死锁处理方法;
- (8) Linux 进程同步和通信机制(自学: 2 学时)

2. 重点、难点

重点: 进程同步与互斥、利用信号量实现进程互斥与同步; 银行家算法;

难点: 利用信号量解决实际进程同步问题。

3. 支撑的课程目标

通过本单元学习, 学生能了解在特定开发环境下利用临界区、信号量、PV 操作等机制实现进程(线程)同步的方法, 能使用经典的同步模型解决实际问题, 具备在软件系统开发中使用进程同步与并发模型优化软件整体性能的能力。学生了解死锁处理的不同策略, 具有使用银行家算法预判系统安全性的能力, 能使用死锁检测算法解除系统死锁, 可部分支撑**课程目标 2**。

(四) 存储管理

1. 主要内容

- (1) 存储器层次及存储器工作原理;
- (2) 存储管理技术, 包括: 连续存储管理、分页存储管理、分段存储管理;
- (3) 虚拟存储管理、虚存页面分配方法和替换策略;

(4) Linux 虚拟存储管理原理、页表机制、缺页异常处理方法。(自学: 2 学时)

2. 重点、难点

重点: 可变分区不同内存分配算法; 分页、分段存储管理地址映射机制; 虚拟存储器常用页面置换算法思想及实现;

难点: 不同内存管理机制、典型分配与置换算法的验证与分析。

3. 支撑的课程目标

通过存储器工作原理、内存管理技术的学习, 学生能理解连续内存管理、页式、段式、分段式管理和虚拟内存管理模型, 掌握逻辑地址到物理地址映射机制, 能分析不同内存管理模型对软件系统效率的影响, 可部分支撑**课程目标 1**。

通过内存页面分配和置换算法及分析的学习, 学生能学会分析不同内存分配、置换算法的效率, 在实际软件系统开发时, 能根据需要设计实现相应算法, 可部分支撑**课程目标 2**。

(五) 设备管理

1. 主要内容

- (1) I/O 系统、I/O 控制方式;
- (2) I/O 软件原理、I/O 中断处理程序、设备驱动程序;
- (3) 单缓冲、双缓冲和多缓冲技术;
- (4) 存储设备的物理结构、磁盘驱动调度技术、磁盘优化技术;
- (5) 设备独立性和设备分配;
- (6) 虚拟设备、SPooling 设计、实现与应用。

2. 重点、难点

重点: 存储设备的物理结构、磁盘驱动调度技术和磁盘的优化技术;

难点: 影响磁盘驱动调度的分析及磁盘的优化分布。

3. 支撑的课程目标

通过 I/O 系统原理、缓冲技术、虚拟设备的学习, 学生能充分理解存储设备的物理结构; 能分析信息存储空间排列方式对软件系统中数据访问效率的影响, 可部分支撑**课程目标 1**。

通过磁盘驱动调度模型和调度算法的学习, 学生能理解不同磁盘调度算法对软件系统数据访问的影响, 学会分析不同调度算法对信息存取效率的影响, 可部分支撑**课程目标 2**。

(六) 文件管理

1. 主要内容

- (1) 文件的基本概念、文件的类型和存取方法;
- (2) 文件目录、目录结构和文件目录检索;
- (3) 文件的组织和数据存储;
- (4) 文件系统功能及实现;

(5) Linux 文件系统结构、VFS 数据结构和文件系统调用实现。(自学: 2 学时)

2. 重点、难点

重点: 文件按名存取实现技术、文件物理存储空间的分配和管理、文件系统调用;

难点: 虚拟文件系统功能分解及模拟, 以及特定软件中利用系统调用实现文件操作。

3. 支撑的课程目标

通过本单元的学习, 学生能掌握文件系统管理的基本功能和实现技术, 能利用文件模拟虚拟内存管理的功能, 具有利用系统调用解决软件设计中与文件有关问题的能力, 可部分支撑**课程目标 1**。

六、实验教学内容及要求

(一) 实验 1 Linux 环境配置及其使用

1. 实验目的

(1) 掌握 Linux 操作系统的安装技术;

(2) 掌握 Linux 操作系统常用基本命令的使用;

(3) 掌握 Linux 环境下简单 C 语言程序开发。

2. 实验内容

(1) 在 windows 操作系统环境下安装虚拟机软件 VMWare, 然后在虚拟机上安装 Linux 操作系统;

(2) 熟悉 Linux 常用基本命令的参数格式及具体使用;

(3) 练习并掌握 Linux 提供的文本编辑器来编辑 C 程序程序, 学会利用 gcc、gdb 编译、调试 C 语言程序。

3. 实验要求

(1) 独立完成实验环境搭建, 使用和理解 Linux 常用命令功能, 观察并简要记录命令执行结果;

(2) 实验结束后提交实验报告, 对实验中遇到的问题、实验结果进行分析总结。

4. 支撑的课程目标

通过本实验, 学生能学会配置虚拟机、在虚拟机环境下安装 Linux 系统, 学会使用 Linux 系统。通过 Linux 使用, 学生可进一步熟悉操作系统功能, 可以更好的从系统软件层面理解复杂软件功能分解方法。通过本实验的训练, 学生也能对虚拟机环境的局限性有深入的理解, 可部分支撑**课程目标 3**。

(二) 实验 2 进程管理

1. 实验目的

(1) 掌握进程的概念, 理解 Linux 系统中进程创建原理, 认识并了解进程并发执行的实质;

(2) 掌握不同的进程创建方法, 熟悉进程的睡眠、同步、撤消等控制方法;

(3) 学会分析进程竞争资源的现象, 掌握解决进程互斥的方法。

2. 实验内容

(1) 编写 C 语言程序：使用系统调用 `fork()` 创建子进程，观察程序运行结果，理解多进程并发运行的特征；

(2) 编写 C 语言程序：使用系统调用 `fork()`、`exec()`、`wait()` 等对进程进行控制，通过观察和分析程序运行结果，理解多进程并发、同步功能；

(3) 编写 C 语言程序：使用系统调用 `fork()`、`lockf()` 等对进程进行控制，通过观察和分析程序运行结果，理解多进程互斥功能。

3. 实验要求

(1) 根据实验内容独立设计满足功能要求的程序，观察和记录进程并发、互斥与同步运行的不同结果，并在实验报告中进行分析；

(2) 实验结束后提交实验报告，对实验中遇到的问题、实验结果进行分析总结。

4. 支撑的课程目标

通过本实验的训练，学生能学会使用 API 实现进程创建、进程同步与互斥，对实验结果进行分析，可以帮助学生更好的理解抽象复杂的多道程序并发原理，使学生具有多线程应用程序开发的基本思维，可部分支撑**课程目标 3**。

(三) 实验 3 进程通信

1. 实验目的

了解 Linux 系统中进程通信几种方法的实现技术，掌握进程间通信方法的不同应用。

2. 实验内容

编写 C 语言程序，分别使用信号机制、管道通信、消息传递通信和共享主存的系统调用，通过观察程序运行结果理解不同进程通信方式的特征。

3. 实验要求

(1) 根据实验内容独立设计满足功能要求的程序，理解信号机制、管道通信、消息传递和共享主存等不同进程通信方式的特征；

(2) 实验结束后提交实验报告，对实验中遇到的问题、实验结果进行分析总结。

4. 支撑的课程目标

通过 Linux 操作系统的系统调用功能实现并发进程间的通信并对实验结果分析，可帮助学生更好的理解和掌握操作系统进程通信机制，帮助学生更好的理解复杂软件系统中不同模块配合完成系统功能的机制，可部分支撑**课程目标 3**。

(四) 实验 4 存储管理

1. 实验目的

(1) 更好的理解虚拟存储管理的特点，掌握请页式存储管理中典型的页面置换算法的基本思想和实现过程，并分析算法的效率；

(2) 掌握文件的基本操作方法。

2. 实验内容

编写 C 语言程序，设计一个虚拟存储区和内存工作区，实现虚拟内存管理中请求页式存储管理的几种基本页面置换算法，并分析内存访问命中率。

3. 实验要求

(1) 利用操作系统提供的文件操作系统调用，用文件模拟存储区，编写程序实现页面置换算法并分析不同算法对内存访问的影响；

(2) 实验结束后提交实验报告，对实验中遇到的问题、实验结果进行分析总结。

4. 支撑的课程目标

通过虚拟存储管理的模拟，可以训练学生分解复杂软件系统功能的能力，通过常用页面置换算法的实现和分析，可以帮助学生更好的理解算法对软件系统效率的影响，可部分支撑课程目标 3。

七、学时分配

课程总学时：64 学时（授课：48 学时，课内实验：16 学时）；学时分配建议见表 2。

课后自学：8 学时。

表 2 教学内容的学时分配

教学内容	授课	实验	自学	课堂测试
操作系统概论	4			
处理器管理	8		2	
进程同步、通信与死锁	11		2	1
存储管理	12		2	
设备管理	6			
文件管理	5		2	1
实验 1 Linux 环境配置及其使用		4		
实验 2 进程管理		4		
实验 3 进程通信		4		
实验 4 存储管理		4		
小计	46	16	8	2

注：授课、实验、课堂测试计入课程总学时，自学不计入课程总学时。

八、考核评价

（一）考核方式

课程目标 1~2 的考核/评价方式包括平时表现和期末考试两种。其中：

平时表现评价依据是课后作业和课堂测验；期末考试采用闭卷考试方式；

课程目标 3 的评价依据是课内实验质量。评价各环节成绩比例见表3。

课程总成绩构成：平时成绩占 30%，实验成绩占 20%，期末考试成绩占50%。

表 3 课程目标评价方式及成绩比例

课程目标	支撑毕业要求 指标点	评价方式及成绩比例（%）			成绩比例 （%）
		平时 （含作业、测验）	实验	期末考试	
课程目标 1	指标点 1.4	15		20	35
课程目标 2	指标点 2.2	15		30	45
课程目标 3	指标点 5.3		20		20
合计		30	20	50	100

注：该表格中比例为课程整体成绩比

（二）考核与评价标准

1. 平时成绩

表 4 平时成绩评价标准

基本要求		优秀 (0.9-1)	良好 (0.8-0.89)	中 (0.7-0.79)	及格 (0.6-0.69)	不及格 (0-0.59)	成绩 比例 (%)
作业	能按时完成作业； 能正确分析题意， 使用相应的原理 或算法分析、解决 问题。	按时完成作业，版 面清晰；能正确使 用算法和原理解决 问题，并得出正确 结论。	按时完成作业，版 面清晰；能正确解 决大部分问题，并 能就部分问题进行 分析。	按时完成作业，能 正确解决大部分 问题。	按时完成作业，能 正确分析大部 分问题并给出 问题解决框架。	不能按时完 成作业或问 题分析错误 较多或认定 为抄袭。	33
课堂 测试	能使用操作系统 知识和基本原理 分析问题；能 根据问题需要设 计算法和模型。	能正确分析问题， 能准确使用算法解 决问题，分析并得 出有效结论。	能准确理解原理并 用于解决大部分问 题，分析结论大部 分有效。	能基于原理给出 解决问题的正确 框架和算法，分析 及结论基本合理。	能给出解决问 题的正确框架， 算法能解决部 分问题，分析基 本合理。	主要原理不 清，算法不 能解决问题。	67

2. 实验成绩

表 5 实验成绩评价标准

基本要求		优秀 (0.9-1)	良好 (0.8-0.89)	中 (0.7-0.79)	及格 (0.6-0.69)	不及格 (0-0.59)	成绩 比例 (%)
实验 平时 成绩	能根据实验目的 和要求设计方案， 按时完成实验并 提交实验报告。	实验方案、运行 结果正确；实验报 告内容完整，并能 总结、分析，得出 有效结论。	实验方案、运行结 果正确；实验报告 内容较完整，但实 验总结分析稍有 不足。	实验方案、实验 结果大部分正 确；能部分总结 实验并得出结 论。	实验方案大部分 正确，能部分实现 程序设计；能部分 总结实验并得出 结论。	大部分实验 要求不能完 成。	40

实验考核成绩	能在 Linux 中根据要求完成设计。	能在规定时间内, 根据要求在 Linux 系统中完成相应程序设计, 功能性能达到要求。	有能力在 Linux 开发和调试软件; 能根据要求设计程序, 大部分功能性能达到要求。	有能力在 Linux 开发和调试软件; 能根据要求给出实现方案, 并大部分实现。	有能力在 Linux 开发和调试软件; 能给出问题的部分解决方案, 并部分实现。	开发和调试软件的能力弱, 无法完成方案设计, 或认定为抄袭。	60
--------	---------------------	---------------------------------------------	---------------------------------------------	------------------------------------------	------------------------------------------	--------------------------------	----

3. 考核成绩

表 6 课程考核成绩评价标准

基本要求		评价标准					成绩比例 (%)
		优秀 (0.9-1)	良好 (0.8-0.89)	中 (0.7-0.79)	及格 (0.6-0.69)	不及格 (0-0.59)	
课程目标 1	能使用操作系统基本原理分析, 并根据要求进行优化和改进。	分析合理, 描述准确; 优化和改进方案合理。	分析合理, 描述大部分准确; 优化和改进方案大部分合理。	分析基本合理; 能提出优化和改进方案框架。	分析基本合理, 能对 60%以上问题提出优化和改进方案框架。	分析错误较多, 无法为大部分问题提出优化和改进方案。	40
课程目标 2	能分析和计算进程调度、死锁处理、内存分配和页面置换、磁盘调度等算法/模型的关键参数。能运用信号量和 PV 操作作为问题设计算法或模型。	计算正确, 设计的算法或模型正确, 能经过分析得到有效结论。	计算正确, 设计的算法或模型基本正确, 能分析但结论有不足。	计算过程正确, 设计的算法或模型框架正确, 分析过程描述大致清楚。	计算过程正确, 设计的算法或模型框架能解决部分问题。	算法使用存在较多错误, 不能得出有效结论。	60

九、推荐教材及参考资料

(一) 推荐教材

1. 费翔林著. 操作系统教程 (第五版). 北京: 高等教育出版社, 2017.

(二) 参考资料

1. 汤小丹. 计算机操作系统. 西安: 西安电子科技大学出版社, 2007.
2. A. S. Tanenbaum 著, 陈向群等译. Modern Operating System. 北京: 机械工业出版社, 2002.
3. William Stallings 著. 操作系统—精髓与设计原理(第五版). 北京: 电子工业出版社, 2006.

执笔人: 王宇英