《操作系统课程设计》

2020级

实验指导书

2023年2月21日

目 录

[《操作系统课程设计》 1](#_Toc97213984)

[1. 课程设计目的 1](#_Toc97213985)

[2. 课程设计任务和要求 1](#_Toc97213986)

[3. 课程设计说明 3](#_Toc97213987)

[3.1 知识点 3](#_Toc97213988)

[3.2 课程设计要点 3](#_Toc97213989)

[3.3 课程设计难点 4](#_Toc97213990)

[4. 课程设计建议 4](#_Toc97213991)

[5. 测试 4](#_Toc97213992)

[6. 设计报告要求 4](#_Toc97213993)

[7. 验收 5](#_Toc97213994)

[**7**.1需要提交的资料（电子版） 5](#_Toc97213995)

[7.2 验收内容及过程 5](#_Toc97213996)

《操作系统课程设计》

## 1. 课程设计目的

* 加深理解操作系统的基本功能、原理和工作机制。
* 理解并掌握操作系统的实现方法和技术。
* 培养学生理解问题、分析问题、解决问题的能力。
* 培养学生团队合作精神、组织协调能力。
* 进一步培养提高学生的编程实践能力。

## 2. 课程设计任务和要求

* 设计并实现一个具有操作系统基本功能的软件。
* 题目：操作系统模拟程序的设计与实现
* 要求该软件
  + 具有操作系统的基本功能：
  + 进程管理功能，如进程创建（new）、进程调度（scheduling）、进程阻塞（block）、进程唤醒（wakeup）、进程同步（synchronize）等。
  + 内存管理功能，进程存储空间的分配和回收等。
  + 文件系统，目录/文件的创建和删除、空间分配和回收
  + 设备管理，设备的申请、分配、使用、释放等
  + UI界面
  + 模拟实现中断机制
* 课程设计基本要求
  + 要求完成的最小功能集合
  + 进程管理和调度
  + 内存管理(存储分配与回收，进程交换)
  + 时钟管理：timer
  + 中断处理：中断响应、中断处理
  + 用图形界面展示多道程序并发执行的过程

可选做部分：

* + 分析LINUX中进程间同步的各种机制
  + 分析LINUX中进程间某种同步机制的实现
  + 分析LINUX中多处理器的调度处理，特别是任务均衡的实现
  + 分析特定典型64位CPU的内存地址空间的多级分页模式及相关的管理
  + 分析LINUX中对大容量磁盘驱动程序的实现（硬件特点）
  + 分析LINUX中典型的64位、多CPU的启动过程（ARM 64位/AMD x86-64位）
  + …
* 课程设计形式
  + 课程设计以开发团队的形式进行
    - 自由组合，每组 5±1 人、设组长一人
    - 每个成员(包括组长在内)都必须至少独立完成一个功能模块的设计与实现、撰写所承担模块的文档。
  + 要求：

（1）每个成员的任务明确，相对独立。

（2）每个成员必须承担一定的功能模块设计开发工作。

（3）整理文档是每个组员任务的一部分。

（4）每个组员完成自己所编写模块的单元测试。

（5）组长统筹安排软件的集成测试。

* 上机环境
  + Windows 或 Linux
  + C/C++ / Java等

## 3. 课程设计说明

### 3.1 知识点

* 进程管理
  + 进程状态（ready, running, waiting, new, terminated）
  + 进程创建、撤销、阻塞、唤醒、同步控制
  + 创建进程时，内存不足，如何处理？
  + swap in/swap out 或者 page in/page out
* 处理机调度
  + 调度算法(FCFS, PRIORITY, RR, ……)
  + 思考：处于swapped ready状态的进程，何时调度？
* 存储管理
  + 存储管理模式：连续分区 / 页式 / 按需调页？
  + 存储空间分配/回收
* 中断机制
  + 中断向量表、中断处理程序
  + 计时器
* 文件系统
  + 目录的逻辑结构（tree型）、物理结构
  + 目录内容（FCB）
  + 目录/文件创建和删除、文件操作（打开/关闭、读/写）
  + 文件数据空间的组织方式（连续、链接、索引），存储空间的分配和回收
  + UI（API，GUI？CLI？）
* 设备管理
  + 设备表、设备状态、设备队列
  + 设备申请、分配、使用、释放
  + API
* 软件界面
  + 手动控制程序的提交执行
  + 动态展示系统运行期间的快照（snapshot），包括：
  + 各并发进程的状态
  + 内存分配情况，显示各进程占用的内存块及其位置、系统空闲内存块及其位置
  + 各设备的状态及设备队列情况

### 3.2 课程设计要点

* 进程
  + 进程结构的定义
  + PCB的定义
  + 进程控制原语的设计（create, stop, block, wakeup, suspend, … ）
  + 进程调度算法的选择与实现
  + 进程的同步与互斥
  + 信号量的定义与操作（ wait, signal ）

可选做部分：

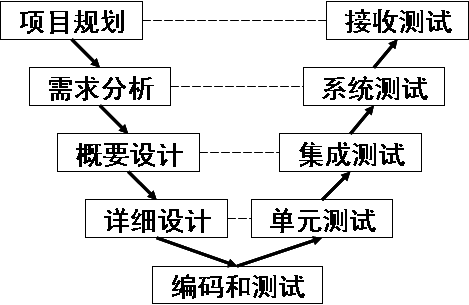
加入生产者-消费者进程到操作系统中，运行进程并产生输出，评估进程可正常执行

* 物理内存管理（分配和回收）
  + 管理模式的选择（连续、页式、按需调页）
  + 交换程序/页面置换算法的选择与实现
* 中断机制与中断处理
  + 中断向量表（键盘中断、磁盘中断、打印机中断）

### 3.3 课程设计难点

* 系统总体设计
* 系统调试

## 4. 课程设计建议

* 明确目标，需求分析（数据流分析、功能分析）
* 总体设计
  + 软件结构设计
  + 模块划分、模块功能
  + 模块之间的关系
  + 模块之间的接口
  + 数据结构定义
  + 用户接口设计
* 详细设计，包括：各模块的详细设计
  + 接口描述（输入、输出）
  + 功能描述
  + 所用数据结构说明
  + 算法描述（程序流程图）
* 编码、调试
* 验收

## 5. 测试

* 单元测试
* 集成测试
* 系统测试
* 接收/验收测试
* 进行软件设计的同时进行测试设计，完成测试计划。
  + 测试环境
  + 测试的功能
  + 测试用例、预期结果
* 测试报告，包括：
  + 测试环境
  + 测试的功能
  + 针对每个功能的测试情况，包括：测试用例、预期的结果、实际测试结果、结果分析
* 在设计测试计划时，不但要考虑正确的测试用例，还要考虑含有异常情况的测试用例。

## 6. 设计报告要求

整理课程设计资料，形成课程设计报告，建议报告包括以下内容：

1．课程设计题目

2．课程设计目标和要求

3．需求分析，包括：数据流图、功能及数据说明等

4．开发环境

5．总体设计说明，包括：

1）数据结构设计

2）总体结构设计：包括

功能模块的划分

模块功能

模块之间的关系

模块之间的接口

3）用户接口设计

6．各部分的详细设计说明，包括：

接口描述

功能描述

所用数据结构说明

算法描述

7．程序清单

注意编程风格，如：

使用有意义的变量名、程序的缩排、程序的内部注释

8．测试报告，包括：

1）测试环境

2）测试计划  
3）针对每个功能的测试情况，包括：测试用例、预期的结果、测试结果及其分析

在设计测试计划时，不但要考虑正确的测试用例，还要考虑含有错误的测试用例。

9．实验总结

1) 实验中遇到或存在的主要问题

2) 改进建议

3) 体会/收获

可选部分：

可选部分LINUX的实现中主要为分析报告。如果进行了其他的一些工作，提供相应的程序的

相关的说明。报告中需要分析相关设计思路，包括相关文件、数据结构，主要算法的说明，并对其特点进行评价。如果进行了一些软件的测试，对测试结果进行分析。

## 7. 验收

### **7**.1需要提交的资料（电子版）

1. 课程设计报告分组表（.doc）
2. 课程设计报告（.doc）
3. 所设计的软件
4. 源代码(带注释)
5. 可运行的程序
6. 测试用例（定义的程序结构）
7. 软件使用说明

* 验收时，携带填写完整并打印好的附件《验收登记表》
* 如发现抄袭，一律按0分计。

### 7.2 验收内容及过程

每组验收时间约40分钟，包括讲解、演示和答辩。

* 讲解Demo的总体功能及软件架构设计、进程的代码结构及指令集。
* 讲解进程控制功能，包括进程的创建和撤销、进程调度算法、进程需要的内存空间的分配和回收策略等。
* 讲解文件系统的实现，以及进程对文件的访问。
* 讲解设备管理功能，以及进程请求设备、释放设备、I/O调度策略等。
* 介绍UI布局设计。
* 运行Demo程序，并演示系统状态的动态变化过程。

要求：每个人讲解自己完成的工作内容，必要时需要讲解所编写的功能代码。

* 查看设计文档