**2018年《操作系统课程设计》选题任务书**

# 题目：可视化仿真实现Linux2.6进程管理与内存管理

# 目的及理论依据

1. **目的**

为了检验自己一学期的操作系统课程的学习情况与掌握程度，以及将该课程知识用编程语言描述的技能，所以本次选择“可视化仿真实现Linux2.6进程管理与内存管理”的题目进行操作系统课程设计。

该课程设计对于理解操作系统中进程管理与内存管理的知识有着重要作用，同时，管理手段采用Linux2.6内核的规则，可借此机会检验自己的编程水平与Linux核心代码的阅读水平。

1. **理论依据**

对于操作系统中涉及到的知识点，本课程设计题目可以对他们很好地进行检验。尤其体现在进程管理与内存管理两方面。

处理器管理是操作系统的重要组成部分，负责管理、调度和分配计算机系统的重要资源——处理器，并控制程序执行。由于处理器管理是操作系统最核心的部分，无论是应用程序还是系统程序，最终都要在处理器上执行以实现其功能，因此处理器管理的优劣直接影响系统性能。程序以进程的形式来占用处理器和系统资源，处理器管理中最重要的是处理器调度，即进程调度，也就是控制、协调进程对处理器的竞争。进程可被调度在一个处理器上交替地执行，不用类型的操作系统可能采取不同的调度策略，交替执行和并行执行都是并发的类型。

进程是操作系统中最重要和最基本的概念之一，引入进程是由系统资源的有限性和系统内的并发性所决定的。进程具有生命周期，由创建而产生，由调度而执行，由撤销而消亡，操作系统的基本功能是进程的创建、管理和撤销。进程是并发程序的执行，是动态的概念，但是为了在处理器上执行仍然需要静态描述。

存储管理是操作系统的重要组成部分，负责管理计算机系统的重要资源——内存储器。由于任何程序和数据必须占用内存空间才能得以执行和处理，因此存储器管理的优劣直接影响系统性能。尽管现代计算机内存容量不断增大，但仍然不能保证有足够大的空间支持大型应用和系统程序及数据的使用。因此，操作系统的主要任务之一是尽可能方便用户使用和提高内存利用率。此外，有效的存储管理也是多道程序设计技术的关键支撑。具体地说，存储管理包含以下功能：存储分配、地址映射、存储保护、存储共享、存储扩充。

# 每位同学拟完成内容

1. **组长**
   1. 三级作业调度过程及算法
   2. 进程死锁检测与撤销算法
   3. 可视化方式呈现过程
   4. 进程同步互斥的实现
2. **组员**
3. CPU部件的仿真
4. 内存空间的仿真实现
5. MMU地址变换
6. 进程与进程原语的设计实现
7. 页表设计实现
8. 页面调度算法、页面分配与回收算法
9. 进程同步互斥的实现

# 三、技术路线图

**1、总体开发环节**

根据选题任务书的安排，由组长负责整个课程设计项目的统筹安排，同时，组长将课程设计的任务进行细分，具体到每一个技术步骤的实现，将某一些任务交由组员完成。当组员完成后，组长对其进行检查确认，并与自己设计的部分进行合并，最终完成本次课程设计的要求。

关键环节如下：

①数据结构的设计。组长与组员共同讨论程序的数据结构设计方法，并最终达成共识。

②裸机部件的模拟。组长与组员共同讨论并确定该程序在设计及运行过程中需要使用到的裸机部件，并将他们抽象成class类结构。同时，对此制定一套使用规范，规定这些类的使用方法及函数接口。

③算法设计。组长与组员共同讨论该程序需要使用的算法，以及如何与逻辑的基本部件进行组合运行。

④时间安排。组长与组员对假期的时间进行规划，安排假期的开发计划，同时，加强交流，积极解决问题。

⑤程序合并，组长与组员将各自按照统一规范编写的代码进行合并，使之成为一个完整的可用的系统。

**2、流程图**

|  |  |  |
| --- | --- | --- |
| **时间** | **组长** | **组员** |
|  | **阶段1** | |
| 1 | 讨论数据结构 | 讨论数据结构 |
| 2 | 确定数据结构 | 确定数据结构 |
| 3 | 硬件类的接口设计 | 硬件类的接口设计 |
| 4 | 硬件类的接口规范设计 | 硬件类的接口规范设计 |
| 5 |  | |
| 6 | **阶段2** | |
| 7 | 可视化界面实现 | CPU部件的仿真 |
| 8 |  | 内存空间的仿真实现 |
| 9 | 三级作业调度过程算法描述 | MMU地址变换 |
| 10 | 进程死锁检测算法描述 | 页表设计实现 |
| 11 | 进程死锁撤销算法描述 | 进程与进程原语的设计实现 |
| 12 | 进程同步互斥算法描述 | 页面调度算法 |
| 13 |  | 页面分配与回收算法 |
| 14 |  | |
| 15 | **阶段3** | |
| 16 | 进程同步互斥算法描述 | 进程同步互斥算法描述 |
| 17 |  | |
| 18 | **阶段4** | |
| 19 | 三级作业调度过程算法实现 | 提供Linux相关知识帮助 |
| 20 | 进程死锁检测算法实现 |
| 21 | 进程死锁撤销算法实现 |
| 22 | 进程同步互斥算法实现 |
| 23 |  | |
| 24 | **阶段5** | |
| 25 | 项目功能测试 | 项目功能测试 |
| 26 | bug检测与修复 | bug检测与修复 |
| 27 | 撰写课程设计报告 | 撰写课程设计报告 |
| 28 | 制作答辩PPT | 制作答辩PPT |
| 29 | 准备答辩 | 准备答辩 |

# 四、关键技术环节

**（1）进程结构设计**

进程结构设计可仿照Linux2.6系统中的进程设计方法，并适应课程设计所需要开发的系统实际需要进行改造，使其充分符合课程设计的要求。

**（2）页面调度、分配、回收算法**

尝试使用linux2.6的二进制伙伴分配算法管理物理页面，Linux2.6为每个管理区使用不同的伙伴系统，内核空间分为三种区，DMA，NORMAL，HIGHMEM，对于每一种区，都有对于的伙伴算法。

linux采用四级分页模型，这四种页表是：页全局目录(PGD)、页上级目录(PUD)、页中间目录(PMD)、页表(PTE)。这里的所有页全局目录、页上级目录、页中间目录、页表，它们的大小都是一个页。页框分配器进行页框的分配，管理区分配器接受动态内存分配与释放的请求，它首先从每CPU页框高速缓存中请求页框，若无法满足才从伙伴系统中请求分配。

页面的使用者可以向PFRA注册回调函数（使用register\_shrink函数）。然后由PFRA在适当的时机来调用这些回调函数，以触发对相应页面或对象的回收。

**（3）三级作业调度过程及算法**

使用Linux的调度器配合CFS算法，根据各个进程的权重分配运行时间。

高级、中级和低级调度作业从提交开始直到完成，往往要经历下述三级调度：

高级调度：(High-Level Scheduling)又称为作业调度，它决定把后备作业调入内存运行；

低级调度：(Low-Level Scheduling)又称为进程调度，它决定把就绪队列的某进程获得CPU；

中级调度：(Intermediate-Level Scheduling)又称为在虚拟存储器中引入，在内、外存对换区进行进程对换。

**（4）线程同步互斥及PV操作实现**

使用linux的信号机制、同步锁与互斥锁来实现PV操作。

Linux中 四种进程或线程同步互斥的控制方法：

1. 临界区:通过对多线程的串行化来访问公共资源或一段代码，速度快，适合控制数据访问。
2. 互斥量:为协调共同对一个共享资源的单独访问而设计的。
3. 信号量:为控制一个具有有限数量用户资源而设计。
4. 事 件:用来通知线程有一些事件已发生，从而启动后继任务的开始。

**（5）进程死锁检测与撤销操作实现**

使用拓扑遍历算法模拟生成有向图进行死锁检测，在发生死锁时按照某种顺序撤销死锁进程，直到死锁状态结束。

# 五、假期开发时间计划

**1、组长**

**2018.01.21 - 2018.01.24：**

* 1. 讨论数据结构
  2. 确定数据结构
  3. 硬件类的接口设计
  4. 硬件类的接口规范设计

**2018.01.25 - 2018.02.01：**

* 1. 可视化界面实现
  2. 三级作业调度过程算法描述
  3. 进程死锁检测算法描述
  4. 进程死锁撤销算法描述
  5. 进程同步互斥算法描述

**2018.02.02 - 2018.02.03：**

1. 进程同步互斥算法描述

**2018.02.11 - 2018.02.20：**

1. 三级作业调度过程算法实现
2. 进程死锁检测算法实现
3. 进程死锁撤销算法实现
4. 进程同步互斥算法实现

**2018.02.21 - 2018.02.24：**

* 1. 项目功能测试
  2. bug检测与修复
  3. 撰写课程设计报告
  4. 制作答辩PPT
  5. 准备答辩

**2、组员**

**2018.01.21 - 2018.01.24：**

* 1. 讨论数据结构
  2. 确定数据结构
  3. 硬件类的接口设计
  4. 硬件类的接口规范设计

**2018.01.25 - 2018.02.01：**

* 1. CPU部件的仿真
  2. 内存空间的仿真实现
  3. MMU地址变换
  4. 页表设计实现
  5. 进程与进程原语的设计实现
  6. 页面调度算法
  7. 页面分配与回收算法

**2018.02.02 - 2018.02.03：**

1. 进程同步互斥算法描述

**2018.02.11 - 2018.02.20：**

1. 提供Linux相关知识帮助
2. 撰写课程设计报告

**2018.02.21 - 2018.02.24：**

* 1. 项目功能测试
  2. bug检测与修复
  3. 撰写课程设计报告
  4. 制作答辩PPT
  5. 准备答辩

# 六、申报成绩

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 班级 | 姓名/学号 | 联系电话 | QQ号 | 申报成绩 | 是否为组长 |
| 网工161班 | 陈扬 19316117 | 17761703297 | 824678119 | A+ | 是 |
| 计科161班 | 梁嘉文 19216126 | 15326099602 | 635229914 | B+ | 否 |