**2018年《操作系统课程设计》选题任务书**

# 题目：可视化仿真实现作业管理与虚页内存管理

# 目的及理论依据

1.1目的

在多任务的作业并发环境下，仿真实现作业及进程并发环境、MMU 地址变换、进程原语、页表生成与虚页调度算法（至少实现 LRU） 、三级作业调度过程及算法（需作业调度到指令集，至少实现三态转换） 、页面分配与回收算法、进程同步互斥、进程死锁检测与撤销算法，并将实现原理过程通过可视化方式呈现，并且要求通过java语言实现相关要求，而且最好涉及整本书的四到五章内容。

1.2理论依据

1.2.1并发

一定时间内，物理机器上有两个或两个以上的程序同处于开始运行但尚未结束的状态，并且次序不是事先确定的。进程的并发性是指一组进程的执行在时间上是重叠的。这里面的并发的实现主要是靠线程来实现的，java里对于线程的操作比较多，比较容易实现。

1.2.3进程原语

原语

操作系统或计算机网络用语范畴。是由若干条指令组成的，用于完成一定功能的一个过程。它是由若干个机器指令构成的完成某种特定功能的一段程序，具有不可分割性，即原语的执行必须是连续的，在执行过程中不允许被中断。这个原语的主要实现方式就是通过调用函数来进行实现。

1.2.4页表

页表是一种特殊的数据结构，放在系统空间的页表区，存放逻辑页与物理页帧的对应关系。每一个进程都拥有一个自己的页表，PCB表中有指针指向页表，实现从页号到物理块号的地址映射。

逻辑地址转换成物理地址的过程是：用页号p去检索页表，从页表中得到该页的物理块号，把它装入物理地址寄存器中。同时，将页内地址d直接送入物理地址寄存器的块内地址字段中。这样，物理地址寄存器中的内容就是由二者拼接成的实际访问内存的地址，从而完成了从逻辑地址到物理地址的转换。页表准备使用列表等结构来实现。

1.2.5进程互斥

由于进程具有独立性和异步性等并发特征，计算机的资源有限，导致了进程之间的资源竞争和共享，也导致了对进程执行过程的制约。进程的互斥由线程相关操作实现。

1.2.6进程间的同步

把异步环境下的一组并发进程因直接制约而互相发送消息而进行互相合作、互相等待，使得各进程按一定的速度执行的过程称为进程间的同步。具有同步关系的一组并发进程称为合作进程，合作进程间相互发送的信号称为消息或事件。

1.2.7死锁

死锁是两个或更多线程阻塞着等待其它处于死锁状态的线程所持有的锁。死锁通常发生在多个线程同时但以不同的顺序请求同一组锁的时候。实现方式考虑的是使用Lock类中的tryLock方法去尝试获取锁，这个方法可以指定一个超时时限，在等待超过该时限之后变回返回一个失败信息。

# 每位同学拟完成内容（分小目标来设计，需要与教材知识点有对应关系）：

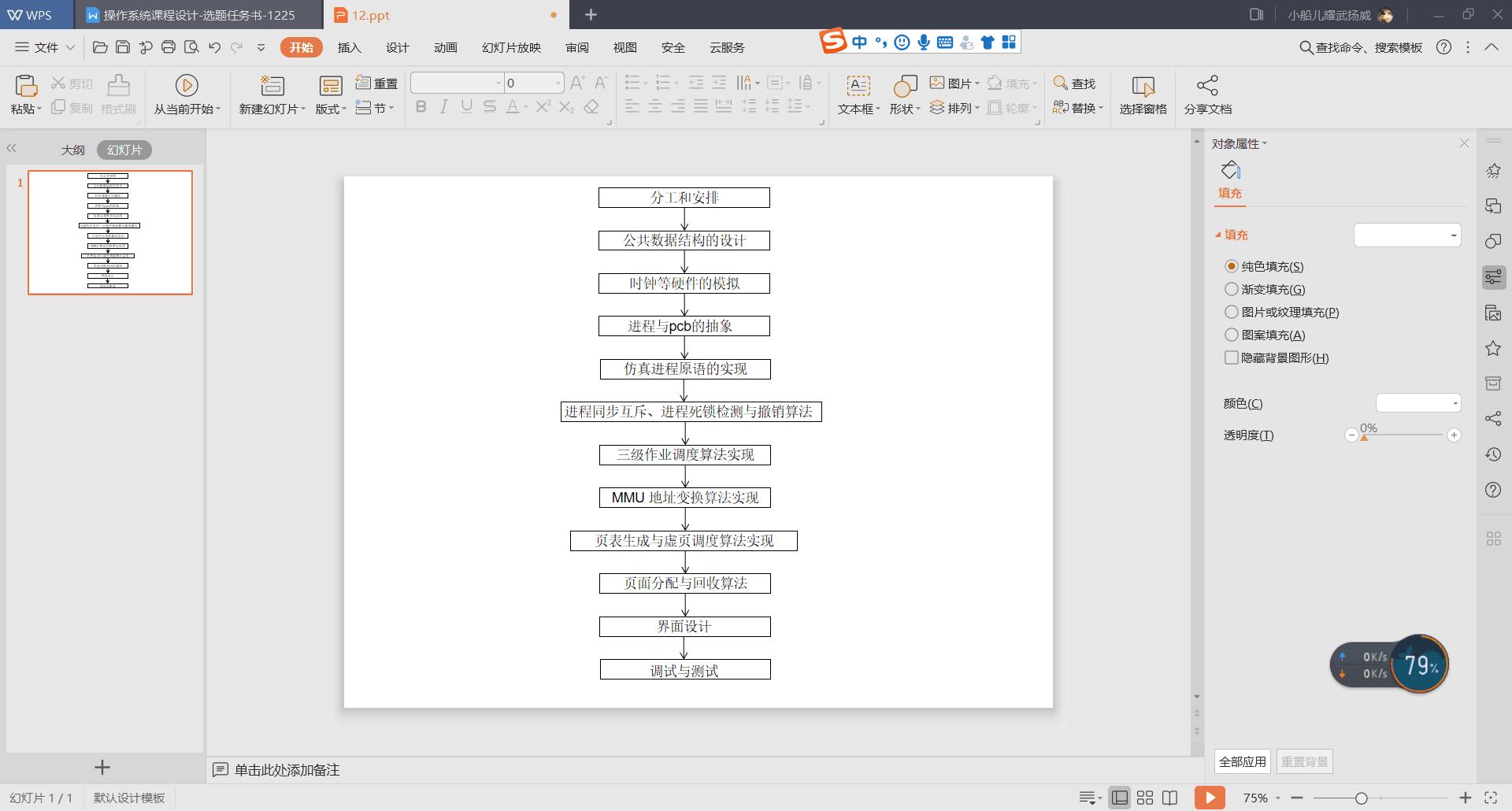
（1）共同完成共同数据结构设计以及时钟等硬件结构的仿真，朱凯同学负责共同数据结构的主要设计，李润隆同学主要的负责时钟等硬件结构的仿真。

（2）李润隆同学负责的是进程原语的编写，三级作业调度过程及算法，进程同步互斥、进程死锁检测与撤销算法，作业及进程并发环境，包括相应的调试测试。

（3）朱凯同学负责的是页面相关的主要操作，包括了调度算法，MMU 地址变换，页表生成与虚页调度算法，页面分配与回收算法，包括相应的测试以及调试。

（4）由于界面设计接口较多，因此最后的界面设计以及综合调试由两个人共同研究完成。

# 三、技术路线图（总体开发环节与流程图）



# 四、关键技术环节

4.1三级作业调度

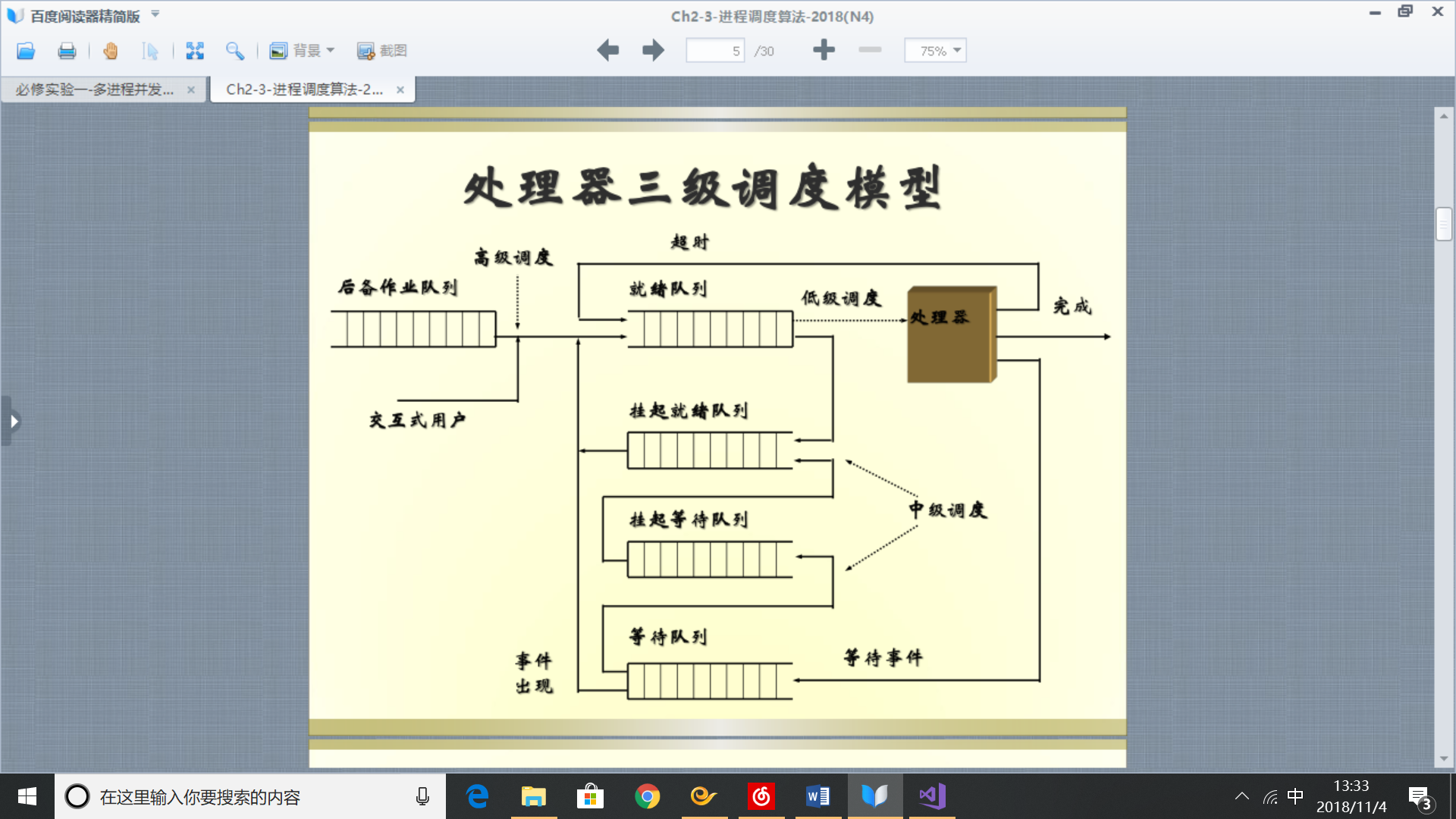
作业从进入系统成为后备作业开始,直到运行结束退出系统为止，需经历不同级别的调度。

•高级调度

•中级调度

•低级调度

三级作业调度具体准备通过一系列原语的创建以及其他调度算法来实现。



4.2 页面调度算法

页式虚拟存储器实现的一个难点是设计页面调度（置换）算法，即将新页面调入内存时，如果内存中所有的物理页都已经分配出去，就要按某种策略来废弃某个页面，将其所占据的物理页释放出来，供新页面使用。本实验准备通过java实现一种到几种常见的页面调度（置换）算法，通过对列表的操作，完成页面的调度。

4.3 MMU地址变换

MMU是Memory Management Unit的缩写，中文名是内存管理单元，它是中央处理器（CPU）中用来管理虚拟存储器、物理存储器的控制线路，同时也负责虚拟地址映射为物理地址，以及提供硬件机制的内存访问授权，多用户多进程操作系统。

4.4 LRU

先进先出调度算法没有考虑页面的使用情况，大多数情况下性能不佳。根据程序执行的局部性特点，程序一旦访问了某些代码和数据，则在一段时间内会经常访问他们，因此最近最少用调度在选择淘汰页面时会考虑页面最近的使用，总是选择在最近一段时间以来最少使用的页面予以淘汰。算法实现时需要为每个页面设置数据结构记录页面自上次访问以来所经历的时间。大概实现方式如下：

利用链表和哈希表。当需要插入新的数据项的时候，如果新数据项在链表中存在（一般称为命中），则把该节点移到链表头部，如果不存在，则新建一个节点，放到链表头部，若缓存满了，则把链表最后一个节点删除即可。在访问数据的时候，如果数据项在链表中存在，则把该节点移到链表头部，否则返回-1。这样一来在链表尾部的节点就是最近最久未访问的数据项。因此，准备通过链表和哈希表来实现具体算法调度。

4.5硬件仿真

时钟仿真：使用一个时间类仿真进行封装。

内存仿真：使用类来进行仿真，一个类可以储存一定空间的数据。CPU仿真：使用CPU类来进行仿真模拟，包含程序计数器等硬件仿真。

磁盘仿真：使用多个文件来进行存储，每个文件再分为磁道等结构，每个文件都可以读出其内容，可以进行读写，进行保存和修改。

MMU：通过类以及类里面相关的构造函数，管理和访问函数进行仿真其结构以及功能。

4.6 数据结构

包括作业控制块设计、作业队列设计、进程控制块设计、进程队列设计、块表设计、页表设计等结构。

物理块设计：与页面大小相等，每个块包括物理块 ID、物理块分配状态、进程 ID等内容。

块表：用类仿真，具体变量有内存总块数、空闲块判定、分配详细信息等。

页表：通过类仿真实现，包含基址表长等信息。

# 五、假期开发时间计划（每位同学分别写，与技术路线图中具体环节有对应关系）

（1）朱凯：

1.18-1.22 公共数据结构的设计

1.23 数据结构以及时钟硬件结构的共同调试

1.24-1.29 MMU地址变换的实现

1.30-2.5上午 页表生成及虚页调度算法的实现

2.8下午-2.11 页面分配算法的实现

2.12-2.14 页面回收算法的实现

2.15-2.19 程序的界面设计

2.20-2.21 界面以及程序的综合调试

2.22-2.25 结题报告等相关文档的撰写

（2）李润隆：

1.18-1.22 时钟等硬件结构的设计

1.23 数据结构以及时钟硬件结构的共同调试

1.24-1.25 进程以及pcb的创建

1.26-1.28 仿真进程原语的创建

1.29-2.1 进程同步互斥

2.2-2.5上午 三级作业调度的部分实现

2.8下午-2.10 三级作业调度算法的实现

2.11-2.14 死锁的检测与撤销算法的实现

2.15-2.19 程序的界面设计

2.20-2.21 界面以及程序的综合调试

2.22-2.25 结题报告等相关文档的撰写

# 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 |
| 计科161 | 李润隆 30216311 | 18795908869 | 1075060506 | B+ | 是 |
| 计科162 | 朱凯 30116306 | 18357522632 | 916786215 | B+ | 否 |

成绩说明：A+：96-100；A：94-95；A-：90-93；B+：86-89；B：84-85；B-：80-83；C+：75-79；C：70-74；D：60-69；E：不及格