

操作系统第一次实训文档

一、实验目的

- 1.加深进程概念的理解，明确进程与程序的区别
- 2.理解操作系统中进程的组织、管理、调度等方法

二、实验内容

编写程序完成单处理器系统的进程调度，要求采用基于时间片多队列反馈式调度策略。具体内容如下所示：

- 1.确定PCB内容及其组织方式
- 2.要求模拟进程创建、就绪、运行、阻塞和完成五个状态
- 3.实现进程创建、进程调度、进程阻塞、进程唤醒和进程撤销五个原语
- 4.模拟进程丛生到灭的完整过程

三、要求和提示

四点要求：

- 1.功能完整：必须模拟5个进程状态（不能只有就绪和运行两个状态！）、能够实现进程创建、调度、阻塞、唤醒和撤销5个功能。
- 2.调度算法：采用“基于可变时间片的反馈多队列调度算法”，即不同优先级（队列）有不同大小的时间片，一次调度执行后没有完成则降级排队。
- 3.参数可变、输入容错：所有参数（个数、类型和格式等）自定，可以通过键盘手工输入，亦可从文件输入，但必须是用户可修改、必须（类型、值域等）容错。
- 4.系统易用、界面友好：必须采用图形用户界面，即进程状态切换要用多窗口（比如功能选择窗口、参数输入窗口和结果输出窗口等）或动画方式展示，必须清晰、直观。
- 5.程序结构清晰：简短清晰的主函数+分层（功能简单）子函数（对象）——体现逐步求精思想。

三点提示：

- 1.如何组织进程：

确定PCB内容：标识信息、状态、优先级、运行时间、I/O时间（时刻与持续时间）和存储地址等信息、现场信息、管理信息。

PCB组织方式：相同状态的进程PCB构成一个队列（即有空闲、就绪、运行、阻塞和完成5个队列）。

- 2.如何创建进程：

申请PCB（从空闲队列）—> 申请资源（模拟）—> 填写PCB—>挂就绪队列

- 3.如何实现处理机调度及进程状态切换：

- 1采用基于可变时间片的多队列反馈式调度策略实现进程调度。

- 2从就绪队列选择一个进程；摘取PCB，挂运行队列；修改状态等PCB内容；保存现场、恢复现场。

- 3模拟运行——可以参考两种场景模拟进程运行：

可以预先设置好各进程的总的运行时间、I/O时间、I/O发生的时刻等信息，之后操作系统控制进程运行，实现状态切换，直到全部进程完成。亦可以采用人工干预方式控制进程状态切换（运行时间已预先设置），比如输入“Esc”进入“阻塞”状态，输入“Enter”则选择（新）进程运行（进程调度），当前进程回到就绪状态;输入“wakeup”,再选择阻塞进程，则被选中的阻塞进程回到就绪状态；输入“finished”，当前进程运行结束，回到完成状态。

4修改（剩余）运行时间：每个优先级队列与一个指定大小的时间片相关联。

四、我的设计思路

技术选型：由于要求用户界面的美观，所以我选择了HTML组织页面，JavaScript组织代码逻辑，使用了Vue.js提高开发效率。

多队列反馈式调度策略主要是指有多个就绪队列，每个就绪队列的优先级不同，优先级越高时间片越小，当进程由于时间片内未完成而回到就绪队列时需要降到下一优先级的就绪队列。使用html+css完成页面布局，使用js控制页面内容、完成核心的逻辑实现。用5个list描述创建、就绪、运行、阻塞、结束队列，使用

```
1  setInterval(function() {  
2  
3  }, time)
```

模拟时间片，每一次调用匿名函数执行一次完整的代码逻辑——依次检查更高就绪队列是否为空，若不为空，调度该队列的第一个进程到运行队列并执行一个时间片，若在该阶段执行完成，则把该进程转移到结束队列并重复执行匿名函数，若没有执行完成，把该进程放入低一优先级的就绪队列等待下一次调度。在代码执行的过程中，可以异步地触发阻塞函数，将运行队列里的进程转移到阻塞队列并设置时间，使用

```
1  setTimeout(function() {  
2  
3  }, time)
```

实现（在time毫秒后执行一次匿名函数——将该进程重新转移到低一优先级的就绪队列）。

完成情况是一周时间内完成，在第二周第一节课接受检查并通过。JavaScript——197行，HTML——274行，CSS——128行，不考虑引入的库共计599行。

主要特点是使用HTML+CSS组织页面UI，利用了当前较为丰富美观的Element-UI，让页面更加好看，用户体验更好。更是加入了适应性布局，丰富用户体验。

可以改进的地方代码可维护性较弱，改进的话可以考虑使用VUEX重构。重新封装核心逻辑，增强代码的可维护性。Element-UI在使用上有些小问题，文档在某些细节上描述的不够具体，导致在CSS上面有一小部分自己的冗余代码，改进的话可以考虑使用其他的CSS框架。