《页面置换算法（下）》微课脚本设计

|  |  |
| --- | --- |
| 本节微课名称 | 页面置换算法（下） |
| 知识点描述 | 页面置换算法的相关原理 |
| 知识点来源 | 《现代操作系统》第三章 第4节 |
| 教学类型 | 讲授-启发-讨论-问答 |
| 设计思路 | 对于大部分学生来说如果单是通过对书本上的内容来对该章节的知识进行学习的话会觉得有些枯燥乏味。而如果是通过看视频的方式，能更灵活直观地表述出需要表达的内容，通过文字、图片和动画特效等，更加能够获取学生的兴趣，从而可以对页面置换算法的知识点有更深入的了解与认知。 |

教学过程：

|  |  |  |  |
| --- | --- | --- | --- |
|  | 内容 | 画面 | 时间 |
| 一、片头 | 老师出场，介绍本节课知识点：   1. 最近最少使用页面置换算法 2. 用软件模拟LRU | 同学们好，今天我给大家带来的是现代操作系统原理课的“页面置换算法（下）”，这节课我们将学习以下内容（列出本次微课的知识点提纲） |  |
| 二、正文讲解 | 1.最近最少使用页面置换算法  引出算法；设计思想；性能好，开销大；动画演示算法的运作过程 | 对最优算法的一个很好的近似是基于这样的观察：在前面几条指令中频繁使用的页面很可能在后面的几条指令中还会被使用。反过来说，已经很久没有使用的页面很有可能在未来较长的一段时间内仍然不会被使用。这个思想提示了一个可实现的算法：最近最少使用页面置换算法（LRU）。（引出算法）  它的设计思想是：选择最后一次访问时间距离当前时间最长的一页并置换，即置换未使用时间最长的一页。（设计思想）  这个算法的性能是最接近最优算法的，虽然可以使用一些特殊硬件来实现这个算法，但是会带来很大的开销。（性能好，开销大）  //下面将介绍最近最少使用算法的一种硬件实现，当然这个方法首先需要硬件的支持。它的思想是：给在页框中的所有页面设置一个矩阵，当某一个页面被访问时，就把这个页面在矩阵中对应的一行设置为1，再把对应的一列设置为0。假设页面访问顺序为：0 1 2 3 2 1 0 3 2 3。当系统访问0号页面时，将对应的一行设置成1，再将对应的一列设置成0。当系统访问1号页面时，将对应的一行设置成1，再将对应的一列设置成0。当所有页面都被设置完成后，如果这时需要进行页面置换，则淘汰在对应的一行中值最小的页面，所以1号页面将被淘汰。//  下面我们通过一个例子来演示最近最少使用页面置换算法的运作过程和计算运作过程中的缺页次数。（动画演示算法的运作过程） |  |
|  | 2.用软件模拟LRU | 最近最少使用算法虽然在理论上可以使用一些特殊硬件来实现，但是只有非常少的计算机拥有这种硬件，而且会带来很大的开销。因此，需要一个能用软件实现的解决方案。（引出用软件模拟LRU）  一种可能的方案称为最不经常使用页面置换算法（NFU）。它的设计思想是：选择访问次数最少的页面进行置换。该算法将每个页面与一个软件计数器相关联，计数器的初值为0。每次时钟中断时，操作系统扫描内存中所有的页面，将每个页面的R位的值加到它的计数器上，这个值可能是0或1。当发生缺页中断时，则置换计数器值最小的页面。（设计思想和具体做法）  最不经常使用算法和最近最少使用算法的思想还是相差甚远的。最近最少使用算法是根据页面的访问时间来判断页面是否淘汰，而最不经常使用算法则是根据页面的访问次数来判断页面是否淘汰。假如有这样一个页面，它的使用次数在所有页面中是最高的，但是这个页面已经在很长的一段时间内没有被使用过了，当发生缺页中断时，就可能会淘汰其他有用的页面而不是这个页面。因此，最不经常使用算法其实并不能很好地模拟最近最少使用算法。 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  | 以上就是本节课的全部内容了，同学们掌握好这节课的知识了吗？下面我们来做几道题巩固一下吧。 |  |
| 三、结尾 | 本节课结束 通过几道小问题回顾今天所学知识 | 转到互动答题 |  |