《页面置换算法（中）》微课教案脚本

|  |  |
| --- | --- |
| 本节微课名称 | 页面置换算法（中） |
| 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 | 教案 |
| 一、片头 | 介绍本节课知识点：   1. 第二次机会页面置换算法 2. 时钟页面置换算法 3. 最近最少使用页面置换算法 | 同学们好，今天我给大家带来的是《现代操作系统》的“页面置换算法（中）”，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. 第二次机会页面置换算法  介绍；设计思想；动画演示算法的运作过程；如果每个页面都被访问过的情况 | 先进先出算法可能会把经常使用的页面置换出去，为了避免这一问题，对该算法做一个简单的修改，由此诞生了第二次机会页面置换算法（SCR）。（介绍）  它的设计思想是：按照先进先出算法选择最老页面，检查其访问位R位，如果为0，则置换该页；如果为1，则给第二次机会，并将R位清0。（设计思想）  下面通过一个例子来演示第二次机会算法的运作过程。可以看到页面A到页面E按照进入内存的时间顺序保存在了链表中。假设在时刻14处发生了一次缺页中断，这时最老的页面是A，它是在0时刻到达的。如果A的R位是0，则将它淘汰出内存；如果A的R位是1，则将A放到链表的尾端，修改它的装入时间为当前时刻（14），并且将R位清0，然后从B页面开始继续搜索合适的页面。（动画演示算法的运作过程）  第二次机会算法就是寻找一个在最近的时钟间隔内没有被访问过的页面。如果所有的页面都被访问过了，该算法就会简化为纯粹的先进先出算法。想象一下，假设在刚刚的例子中所有页面的R位都被设置成1了，操作系统将会一个接一个地把每个页面都移动到链表的尾部并清除被移动的页面的R位。最后算法又将回到页面A，此时它的R位已经被移除了，因此A页面将被淘汰，所以这个算法总是可以结束的。（如果每个页面都被访问过的情况） |
|  | 1. 时钟页面置换算法   引出算法；设计思想；原理；动画演示算法的运作过程 | 尽管第二次机会算法是一个比较合理的算法，但它经常需要在链表中移动页面，既降低了效率又不是很有必要。（引出算法）  一个更好的办法是利用时钟页面置换算法（CLOCK），它的设计思想是：把所有的页面保存在一个类似钟面的环形链表中，一个表针指向最老的页面，然后检查这个页面的R位，并根据R位采取动作。（设计思想）  当发生缺页中断时，算法首先检查表针指向的页面，如果它的R位是0就淘汰该页面，并把新的页面插入这个位置，然后把表针前移一个位置；如果R位是1就清除R位并把表针前移一个位置。重复这个过程直到找到了一个R位为0的页面为止。（原理）  下们通过一个例子来演示时钟页面置换算法的运作过程。    假设当前表针指向页面A，它的R位是1，清除R位并把表针前移一个位置。  此时表针指向页面B，它的R位也是1，清除R位并把表针前移一个位置。  此时表针指向页面C，它的R位是0，所以淘汰页面C并把新的页面插入这个位置，然后把表针前移一个位置。下一次缺页中断时，将从页面D开始检查。（动画演示算法的运作过程） |
|  | 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；算法的一种硬件实现 | 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）  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。    系统给某进程分配3个页框，初始为空。  进程执行时，页面走向序列为：2 3 2 1 5 2 4 5 3 2 5 2  系统首先访问2号页面，缺页一次，将2号页面调入页框。  然后访问3号页面，缺页一次，将3号页面调入页框。  然后访问2号页面，没有缺页，无需操作。  然后访问1号页面，缺页一次，将1号页面调入页框。  然后访问5号页面，缺页一次，由于页框中3号页面最后一次访问时间距离当前时间最长，所以淘汰3号页面并将5号页面调入页框。  然后访问2号页面，没有缺页，无需操作。  然后访问4号页面，缺页一次，由于页框中1号页面最后一次访问时间距离当前时间最长，所以淘汰1号页面并将4号页面调入页框。  然后访问5号页面，没有缺页，无需操作。  然后访问3号页面，缺页一次，由于页框中2号页面最后一次访问时间距离当前时间最长，所以淘汰2号页面并将3号页面调入页框。  然后访问2号页面，缺页一次，由于页框中4号页面最后一次访问时间距离当前时间最长，所以淘汰4号页面并将2号页面调入页框。  然后访问5号页面，没有缺页，无需操作。  然后访问2号页面，没有缺页，无需操作。  所以进程运用最近最少使用页面置换算法在运作过程中一共有7次缺页。（动画演示算法的运作过程）  这个算法的性能虽然是最接近最优算法的，但是实现起来会带来很大的开销。为了完全实现这个算法，需要在内存中维护一个所有页面的链表，最近最多使用的页面放在表头，最近最少使用的页面放在表尾。困难的是在每次访问内存时都必须要更新整个链表。在链表中找到一个页面，删除它，然后把它移动到表头是一个非常费时的操作，即使使用硬件实现也一样费时。（性能好，开销大）  然而，还是有一些使用特殊硬件实现这个算法的方法。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，这个页面就是最近最少使用的页面。所以1号页面将被淘汰。（算法的一种硬件实现） |
| 三、结尾 | 结束语 | 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。（转到互动答题） |