《页面置换算法（上）》微课脚本设计

|  |  |
| --- | --- |
| 本节微课名称 | 页面置换算法（上） |
| 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. 页面置换 2. 最优页面置换算法 3. 最近未使用页面置换算法 4. 先进先出页面置换算法 5. 第二次机会页面置换算法 6. 时钟页面置换算法 7. 时钟算法实现最近未使用算法 8. 最近最少使用页面置换算法 | 大家好，……。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（上）”，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.页面置换  为什么要进行页面置换；页面置换是什么；换出的页面去哪了；引出页面置换算法  缺页中断：1、如果内存有空闲，就直接将页面从磁盘调入；2、内存已满，就需要先进行页面置换 | 首先我们需要知道，为什么要进行页面置换，页面置换又是什么呢？  当系统发生缺页中断时，如果此时内存有空闲，就可以直接将页面从磁盘调入；但如果此时内存已满，就需要进行页面置换。（为什么要进行页面置换）  页面置换指的是，操作系统必须在内存中选择一个页面将其换出内存，以便为即将调入的页面腾出空间。（页面置换是什么）  那么被换出的页面会去到哪里呢？如果这个页面在内存驻留期间已经被修改过，就必须把它写回磁盘以更新该页面在磁盘上的副本；如果该页面没有被修改过，那么它在磁盘上的副本已经是最新的，不需要回写，直接用调入的页面覆盖被淘汰的页面就可以了。（换出的页面去哪了）  那么我们应该选择内存中的哪一个页面来置换呢？虽然可以随机地选择一个页面，如果每次都选择不常使用的页面会提升系统的性能，但是如果一个频繁使用的页面被置换出内存，很可能它在很短时间内又要被调入内存，这会带来不必要的开销。所以这时就要用到所谓的页面置换算法了，下面将介绍几个最重要的页面置换算法。（引出页面置换算法） |  |
|  | 2.最优页面置换算法  介绍；设计思想和原理；能否实现，如何实现；算法的意义；动画演示算法的运作过程 | 页面置换算法又称页面淘汰算法或页面替换算法。接下来要介绍的是最优页面置换算法（OPT）。顾名思义，这个算法应该是最优的。（介绍）  它的设计思想是：置换以后不再需要的或很远的将来才会用到的页面。最优页面置换算法规定应该置换标记最大的页面。如果一个页面在800万条指令内不会被使用，另外一个页面在600万条指令内不会被使用，则置换前一个页面，从而把因需要调入这个页面而发生的缺页中断推迟到将来，越久越好。（设计思想和原理）  但是，这个算法能实现吗？我们怎么知道各个页面下一次将在什么时候被访问呢？因此，这个算法的实现，是建立在已经知道页面走向序列的基础之上的。一个程序运行之后，如果我们把所有页面的访问情况记录下来，就可以针对记录的结果，运用最优页面置换算法，来给出运行过程中的哪些页面需要被换出内存。（能否实现，如何实现）  这个算法最大的意义，是它能作为一种标准来衡量其他可实现的页面置换算法的性能之好坏，性能最接近最优页面置换算法的则称之为好的算法，这就是这个算法最主要的作用。（算法的意义）  下面我们通过一个例子来演示最优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）    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1号页面在之后不再被需要，所以淘汰1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2号页面在页框中的3个页面中最晚被需要，所以淘汰2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，没有缺页，无需操作。  然后访问2号页面，缺页一次，由于4号页面在之后不再被需要，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。  所以进程运用最优页面置换算法在运作过程中一共有6次缺页。 |  |
|  | 3.最近未使用页面置换算法  引出可以实际使用的页面置换算法；两个状态位的介绍；设计思想、具体做法；启动进程后两个状态位的情况；根据两个状态位的值分成4类页面；算法的实现 | 最优页面置换算法虽然对评价其他算法很有用，但它在实际系统中却不能使用。下面将研究可以在实际系统中使用的算法。（引出可以实际使用的算法）  在大部分具有虚拟内存的计算机中，系统为每一个页面设置了两个状态位。当页面被访问时设置R位为1，且R位会被定期清零（复位）；当页面被修改时设置M位为1，M位不会被定期清零（复位），原因是在决定一个页面是否需要写回磁盘时将用到这个信息。这些位包含在每个页表项中。一旦设置某位为1，它就一直保持1直到操作系统将它清零（复位）。（两个状态位的介绍）  接下来要介绍的是最近未使用页面置换算法（NRU），它的设计思想是：选择在最近一段时间内未使用过的一页进行置换。具体做法是根据页表项的两位，来决定哪个页面要被置换。（设计思想、具体做法）  当发生缺页中断时，操作系统检查所有页面的R位和M位，并把它们分为4类：  第1类：无访问，无修改（R=0，M=0）  第2类：无访问，有修改（R=0，M=1）  第3类：有访问，无修改（R=1，M=0）  第4类：有访问，有修改（R=1，M=1）（根据两个状态位的值分成4类页面）  而最近未使用算法随机地从以上4类编号最小的非空类中挑选一个页面淘汰。（算法的实现） |  |
|  | 4.先进先出页面置换算法  设计思想；如何实现；用超市的例子来解释；算法的弊端；动画演示算法的运作过程 | 接下来要介绍的是先进先出页面置换算法（FIFO），它的设计思想是：选择在内存中驻留时间最长的页面来进行置换。（设计思想）  该算法可以通过页面链表法来实现，每加载一个页面到内存，就将其挂入到链表，最早进入的页面放在表头，最新进入的页面放在表尾。当发生缺页中断时，淘汰表头的页面，并将新调入的页面加到表尾。（如何实现）  这个算法也存在一定的弊端。设想有一个超市，超市里的货架已经摆满了商品，这时有一批新的商品进来，需要淘汰旧的商品来替换新的商品。如果按照先进先出算法，淘汰最早一批进入超市的商品来替换新的商品，就会出现一个问题，如果这批旧的商品是人们经常会用到的，比如矿泉水或者酱油，那短时间内这批商品还是会重新进入超市，这就会带来不必要的开销。（用超市的例子来解释）  而这个算法用于页面置换也是一样的，如果最早进入内存的页面是经常使用的，当这个页面被置换出去后，短时间内还是会被调入内存。由于这一原因，很少会使用纯粹的先进先出算法。（算法的弊端）  下面我们通过一个例子来演示先进先出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）    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2号页面驻留时间最长，所以淘汰2号页面并将5号页面调入页框。  然后访问2号页面，缺页一次，由于页框中3号页面驻留时间最长，所以淘汰3号页面并将2号页面调入页框。  然后访问4号页面，缺页一次，由于页框中1号页面驻留时间最长，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5号页面驻留时间最长，所以淘汰5号页面并将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5号页面，缺页一次，由于页框中2号页面驻留时间最长，所以淘汰2号页面并将5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号页面驻留时间最长，所以淘汰4号页面并将2号页面调入页框。  所以进程运用最优页面置换算法在运作过程中一共有9次缺页。 |  |
|  | 5.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） |  |
|  | 6.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）  下面我们通过一个例子来演示时钟页面置换算法的运作过程。（动画演示算法的运作过程） |  |
|  | 7.时钟算法实现最近未使用算法  运作过程 | 同样我们也可以利用时钟算法来实现最近未使用算法，具体运作过程是这样的：1、从指针的当前位置开始，扫描页框缓存区，选择遇到的第一个（R=0,M=0）页框用于置换（本次扫描过程中，对R位不做任何修改）   1. 如果第一步失败，则重新扫描，选择第一个（R=0,M=1）页框（本次扫描过程中，对每个跳过的页框，将其R位设置成0） 2. 如果第二步失败，指针将回到它最初的位置，并且集合中所有页框的R位均为0。重复第一步，并且如果有必要再重复第二步。最后将可以找到供置换的页框。（运作过程） |  |
|  | 8.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）  这个算法的性能虽然是最接近最优算法的，但是想要实现这个算法会带来很大开销。（性能好，开销大）  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） |  |
|  |  | 老师发出问题互动“？”  等待一段时间后给出回答 |  |
| 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 |  |