塞,才会继续从队列中选择第一个进程接着运行
FCFS对长作业有利,适用于CPU繁忙型作业系
统,而不适用I/O繁忙型的系统 优先选择运行时间最短的进程来运行,有助于提
高系统吞吐量
SJF对长作业不利,很容易造成一种极端现象
权衡了短作业和长作业,每次进行进程调度时, 先计算「响应比优先级」,然后把「响应比优先 级」最高的进程投入运行
优先级=(等待时间+要求服务时间)/要求服务时间
如果两个进程的「等待时间」相同,「要求的服
务时间」越短,「响应比」就越高,短作业进行 容易被选中运行
如果两个进程「要求的服务时间」相同时,「等 待时间」越长,「响应比」就越高,兼顾到了长
作业进程,等待时间足够长响应比提升
最古老、最简单、最公平的调度算法 每个进程被分配一个时间段,称为时间片(
, RR) Quantum),即允许该进程在该时间段中运行
如果时间片设置的太短会导致过多的进程上下文 切换,降低了CPU效率;反之太长会对短作业的 响应时间边长(通常20ms-50ms)
从就绪队列中选择最高优先级的进程进行运行
创建进程的时候,已经确定了优先级,整个静态优先级 运行时间优先级都不会变化
动态优先级 根据进程的动态变化调成优先级
就绪队列出现优先级高的进行,运行完当前进程
非抢占式 再选择优先级高的进程 当前进程法
当就绪队列出现优先级高的进程,当前进程挂 抢占式 起,调度优先级高的进程运行
可能会导致优先级低的进程永远不会运行
「时间片轮转算法」和「最高优先级算法」综合发展 「多级」表示有多个队列,每个队列优先级从高到低
名同事优先级越高时间片越短
「反馈」表示如果有新的进程加入优先级高的队列,立刻 停止当前正在运行的进程,转而去运行优先级高的队列 设置多个对列。赋予每个队列不同的优先级。每
设置多个对列,赋予每个队列不同的优先级,每 个「对列优先级从高到低」,同时「优先级越高 时间片越短」
新的进程会被放到第一级队列的末尾,按先来先 服务的原则排队等待被调度,如果在第一级队列 规定的时间片没有运行完成,则转入第二级对列 工作流程 尾部,以此类推,直至完成
当较高优先级的队列为空,才调度较低优先级的对 列中的进程运行。如果进程运行时,有新进程进入
较高优先级的队列,则停止当前运行的进程将其移 入到原队列末尾,接着让较高优先级的进程运行
该算法很好的兼顾了长短作业,同时又较好的响应时间
置换在「未来」最长时间不访问的页面
算法实现需要计算内存中每个逻辑页面「下一 次」访问时间,然后比较,选择未来最长时间不
访问的页面
实际系统中无法实现,因为访问时动态的,无法 预知的,所以页面置换算法是为了衡量你的算法 效率,如果越接近说明效率越高
预知的,所以页面置换算法是为了衡量你的算法 效率,如果越接近说明效率越高
预知的,所以页面置换算法是为了衡量你的算法 效率,如果越接近说明效率越高 选择在内存驻留时间很长的页面进行置换
预知的,所以页面置换算法是为了衡量你的算法效率,如果越接近说明效率越高 选择在内存驻留时间很长的页面进行置换 选择最长时间没有被访问的页面进行置换 跟LRU近似,对FIFO的一种改进
预知的,所以页面置换算法是为了衡量你的算法效率,如果越接近说明效率越高 选择在内存驻留时间很长的页面进行置换 选择最长时间没有被访问的页面进行置换 跟LRU近似,对FIFO的一种改进
预知的,所以页面置换算法是为了衡量你的算法 效率,如果越接近说明效率越高 选择在内存驻留时间很长的页面进行置换 选择最长时间没有被访问的页面进行置换 跟LRU近似,对FIFO的一种改进
预知的,所以页面置换算法是为了衡量你的算法 效率,如果越接近说明效率越高 选择最长时间没有被访问的页面进行置换 跟LRU近似,对FIFO的一种改进 将所有页面保存在类似钟面的「环形链表」,一 个指针指向最老的页面,当发生中断的时候,先 检查表指针的页面 如果访问的位置是O就淘汰该页面,并把新页面 插入到这个位置,然后指针后移 如果访问位是1就清除访问位,把表针前移一个位置,重复这个过程找到访问位为0的页面位置
预知的,所以页面置换算法是为了衡量你的算法 效率,如果越接近说明效率越高 选择最长时间没有被访问的页面进行置换 跟LRU近似,对FIFO的一种改进 将所有页面保存在类似钟面的「环形链表」,一 个指针指向最老的页面,当发生中断的时候,先 检查表指针的页面 如果访问的位置是0就淘汰该页面,并把新页面 插入到这个位置,然后指针后移
预知的,所以页面置换算法是为了衡量你的算法 效率,如果越接近说明效率越高 选择最长时间没有被访问的页面进行置换 跟LRU近似,对FIFO的一种改进 将所有页面保存在类似钟面的「环形链表」,一 个指针指向最老的页面,当发生中断的时候,先 检查表指针的页面 如果访问的位置是O就淘汰该页面,并把新页面 插入到这个位置,然后指针后移 如果访问位是I就清除访问位,把表针前移一个位置,重复这个过程找到访问位为O的页面位置 发生缺页中断时,选择「访问次 数」最少的页面,将其淘汰 需要增加一个计数器实现,硬件成本较高,另外对这个计数器直找那个页面访问次数最小,直找链表本
预知的,所以页面置换算法是为了衡量你的算法 效率,如果越接近说明效率越高 选择最长时间没有被访问的页面进行置换
例如的,所以页面置換質法是为了衡量你的算法 效率,如果越接近说明效率越高 选择最长时间没有被访问的页面进行置换 跟LRU近似,对FIFO的一种改进 将所有页面保存在类似钟面的「环形链表」,一 个指针指向最老的页面,当发生中断的时候,先 检查表指针的页面 数生中断的,选择「访问次数量小,查找链表本身,如果链表长度很大,是非常耗时,效率不高 LFU只考虑了频率问题,没考虑时间问题,如果 有些页面而过去的时间访问频率很高,但是现在 已经没有访问了,而当前访问的页面由于没有这 些页面访问次数高,发生缺项中断时,可能会误
预知的,所以页面置接衡法是为了衡量你的算法效率,如果越接近说明效率越高 选择在内存驻留时间很长的页面进行置换
例如的,所以页面置換質法是为了衡量你的算法 效率,如果越接近说明效率越高 选择最长时间没有被访问的页面进行置换 跟LRU近似,对FIFO的一种改进 将所有页面保存在类似钟面的「环形链表」,一 个指针指向最老的页面,当发生中断的时候,先 检查表指针的页面 数生中断的,选择「访问次数量小,查找链表本身,如果链表长度很大,是非常耗时,效率不高 LFU只考虑了频率问题,没考虑时间问题,如果 有些页面而过去的时间访问频率很高,但是现在 已经没有访问了,而当前访问的页面由于没有这 些页面访问次数高,发生缺项中断时,可能会误
预知的,所以页面置换算法是为了衡量你的算法效率,如果越接近说明效率越高 选择在内存驻留时间很长的页面进行置换 选择最长时间没有被访问的页面进行置换 跟LRU近似,对FIFO的一种改进 将所有页面保存在类似钟面的「环形链表」,一 个指针指向最老的页面,当发生中断的时候,先 检查表指针的页面 发生缺页中断时,选择「访问次数」最少的页面,将其淘汰 需要增加一个计数器实现,硬件成本较高,另外对这个计数器查找那个页面访问次数最小,直按链表本身,如果链表长度很大,是非常耗时,效率不高 LEU只考虑了频率问题,设考虑时间问题,如果 有些页面面问过去的时间访问频率很高,但是现在 已经没有访问了,而是前访问的页面由于没有这 些页面面问过去的时间访问频率很高,但是现在 已经没有访问了,而能会误 仿当前开始频繁访问,但访问次数还不高的页面
预知的,所以页面置换箅法是为了衡量你的算法 效率,如果越接近说明效率越高 选择在内存驻留时间很长的页面进行置换 选择最长时间没有被访问的页面进行置换 跟LRU近似,对FIFO的一种改进 将所有页面保存在类似钟面的「环形链表」,一 个指针指向最老的页面,当发生中断的时候,先 检查表指针的页面 发生缺页中断时,选择「访问次数」是少的页面,将其淘汰 需要增加一个计数器实现,硬件成本较高,另外对这个计数器直找那个页面访问次数是小,直找链表本身,如果链表长度很大,是非常耗时,效率不高 IFU只考虑了频率问题,没考虑时间问题,如果有些页面所过去的时间访问频率很高,但是现在 已经没有访问了,而当前访问的页面由于没有这些页面前内次数高,发生缺页中断时,可能会误 伤当前开始频繁访问,但访问次数还不高的页面 先到来的请求先被服务 优先选择从当前磁头位置所需寻
预知的,所以页面置换解法是为了衡量你的算法 效率,如果越接近说明效率越高 选择在内存驻留时间很长的页面进行置换 选择最长时间没有被访问的页面进行置换 跟LRU近似,对FIFO的一种改进 将所有页面保存在类似钟面的「环形链表」,一个指针指向最老的页面,当发生中断的时候,先检查表指针的页面 超大型这个位置,然后指针后移 如果访问的位置是O就淘汰该页面,并把新页面插入到这个位置,然后指针后移 如果访问的位置是O就淘汰该页面,并把新页面插入到这个位置,然后指针后移 如果访问位是可能清除访问位,把表针前移一个位置,重复这个过程找到访问位为O的页面位置 发生缺页中断时,选择「访问次数」最少的页面,将其淘汰 需要增加一个计数器类别,硬件成本较高,另外对这个计数器重找那个页面访问次数最小,直线链表本身,如果链表长度被大,是非常耗时,效率不高 LFU只考虑了频率问题,没考虑时间问题,如果有些页面再过去的时间访问频率很高,但是现在已经没有访问了,而当前访问的页面由于没有这些页面访问了,而当前访问的页面由于没有这些页面访问了,而当前访问的页面由于没有这些页面访问了,而当前访问的页面由于没有这些页面访问了,而当前访问的页面的于没有这些页面访问方面,没考虑时间问题,如果
预知的,所以页面置换算法是为了衡量你的算法效率,如果越接近说明效率越高 选择程代时间没有被访问的页面进行置换
類如即,所以页面置換質法是为了衡量你的算法 效率,如果越接近说明效率越高
類如此,所以页面置換資源是由了衡量你的算法 效率,如果越接近说明效率越高 跟LRU近似,对FIFO的一种改进 据LRU近似,对FIFO的一种改进 将所有页面保存在类似钟面的「环形结表」,一 个情针指向最老的页面,当发生中断的时候,先 检查表描针的页面 发生缺页中断时,选择「访问次数 最少的页面,将其输法 需要增加一个计数器实现,硬件成本较高,另外对这个计数器实现,硬件成本较高,另外对这个计数器实现,硬件成本较高,另外对这个计数器实现,硬件成本较高,另外对这个计数器实现,硬件成本较高,另外对这个计数器实现,硬件成本较高,另外对这个计数器实现,硬件成本较高,是处较可由的可以数量小,查找结束本身,如果转形可远的问题,如果不高 IPU尺考虑了频率的时间问题,如果 有些页面而记录的。 发生较可由时,可能会误 伤当前开外频繁访问,但访问次数还不高的页面 先到来的请求先被服务 优先选择从当前磁头位置所需寻 道是现在已经没有访问了,而当前访问的页面由于没有这些页面而记录数于不同,是一段不同时,可能会误 伤当前开始频繁访问,但访问次数还不高的页面 先到来的请求先被服务 优先选择从当前磁头位置所需寻 道是现的时间请求 可能出现「饥饿现象」 显寻道时间可能会出现的饥饿问题 一个方向上移动,访问所有未完成的请求, 头型认为的所有未完成的请求,
孫如於,所以页面徵數算法是为了後屬你的算法 效率,如果越接近说明效率越高 选择在内存驻留时间很长的页面进行置换
预知的,所以页面的跨量法是为了德量你的剪法 效率,如果越按近邻的欢座越高 选择在内存驻留时间很长的页面进行置换 选择程长时间设有被访问的页面进行置换 提LRU近似,对FIFO的一种改进 郑所有页面保存在类似种面的「环形稳表」,一 个部叶指向最老的页面,当发生中新的时候,先 检查表指针的页面 发生缺项中新时,选择「访问次数」最少的页面,将其淘汰 需要描加一个计数器实现,逐件成本较高,另外对这个计数器实现,还是不能到,实现单位表达的可以为的页面的可以发展小,更对脑表本身,如果随来投度大,是非常耗时,效率不高 1FU只尽多了新年间题,没考虑时间问题,如果有些方面的对话的时间面由于没能这 生现面的可以发起小,更加速和不是 已经经验访问了,而当前访问的项面由于没能这 生现两面的对象。发生缺项中断时,可除会误 伤当前开始频繁初间,但如可次数还不高的页面 不利度和多种,或是不能被多 优先选择从当前磁头位置所需寻面最短的的间离来 可能出现「饥饿现象」 可能出现,可以使用

优化循环扫描算法,不需要移动到最始端或最末端,仅仅移动到最远请求位置然后反向移动,「 C-LOOK 反向移动过程中不会响应请求」

LOOK和C-LOOK算法

「非抢占式」每次从就绪对列选择最先进入对列的进程,然后一直运行,直到进程退出或被阻