

第三章-内存管理

虚拟内存管理

请求分页管理方式

页表项：页号（隐含）+物理块号+状态位+访问字段（置换算法）+修改位（写策略）+外存地址

进程的页框分配

驻留集大小：各一个进程分配的页框集合就是这个进程的驻留集，反映了进程得到的内存空间，驻留集太小导致缺页太多，太大导致进程并发度降低

内存分配策略

- 固定分配局部置换：保持分配的内存不变，缺页置换
- 可变分配全局置换：可以适当增加或减小分配内存，缺页会适当的增加分配内存
- 可变分配局部置换：只有当缺页率过高或过低才改变内存分配，保持缺页率正常水平

物理块调入算法

- 平均分配算法：所有物理块平均分给进程
- 按比例分配算法：按进程大小比例分配物理块
- 优先权分配算法：部分物理块按比例，部分物理块按优先权

调入页面时机

- 预调页策略：根据局部性原理，一次调入若干相邻的页面会更加有效
- 请求调页策略：需要访问的页面不在内存时请求调页，调入的页一定会被访问，主要使用

从何处调页

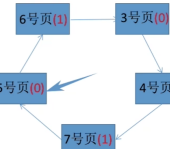
- 外存分为：文件区（离散分配）和对换区（连续分配，所以对换更快）
- 系统有足够的对换区空间：运行前将进程有关的文件复制到对换区，全部从对换区调入
- 系统缺少足够的对换区空间：不会被修改的文件从文件区调入，换出后不会因为修改而反复出入；会被修改的文件从对换区调入
- UNIX方式：未运行过的文件从文件区调入，运行过被换出的文件放在对换区

页面置换算法

OPT最佳置换算法（理想情况）：淘汰最长时间内不再访问的页面

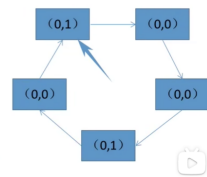
FIFO先进先出算法，会出现Belady异常：增加物理块缺页次数反而增加，队列式算法

LRU最近最久未使用：算法性能较好，堆栈类算法需要寄存器和栈的硬件支持



CLOCK时钟置换算法
访问位每次访问到置为1，指针循环扫描，扫过一周后将所有访问位置为0，缺页时替换第一个扫描到的访问位为0的页面

算法规则：将所有可能被置换的页面排成一个循环队列
第一轮：从当前位置开始扫描到第一个(0,0)的帧用于替换。本轮扫描不修改任何标志位
第二轮：若第一轮扫描失败，则重新扫描，查找第一个(0,1)的帧用于替换。本轮将所有扫描过的帧访问位设为0
第三轮：若第二轮扫描失败，则重新扫描，查找第一个(0,0)的帧用于替换。本轮扫描不修改任何标志位
第四轮：若第三轮扫描失败，则重新扫描，查找第一个(0,1)的帧用于替换。
由于第二轮已将所有帧的访问位设为0，因此经过第三轮、第四轮扫描一定会有一个帧被选中，因此改进型CLOCK置换算法选择一个淘汰页面最多会进行四轮扫描



改进型的CLOCK算法
第一优先：没访问没修改
第二优先：没访问有修改
第三优先：有访问没修改
第四优先：有访问有修改

虚拟内存管理分析

抖动和工作集

抖动：刚换出的页面马上又要换入内存，这种频繁的调度行为称为抖动或者颠簸，根本原因是分配给每个进程的物理块（驻留集）太少

工作集：某段时间内进程要访问的页面集合，驻留集的大小不能小于工作集，否则会出现抖动现象

内存映射文件

操作系统提供的一个系统调用返回一个文件的虚拟指针用于操作，用于将一个文件映射到进程的虚拟地址空间中，这样进程就用访问主存的方式读写文件，进程关闭文件时，由操作系统将其写回磁盘

多个进程可以映射同一个文件方便共享；程序员编写程序更简单，文件的IO部分由操作系统完成优化；

虚拟存储器性能影响因素

对于分配给进程的物理块个数，只需要保证缺页率保持在一个很低的范围即可

关于写回磁盘的频率，系统设置了一个已修改换出页面的链表，用于存储被换出的页面，等到一定数量时再一起写回磁盘，可以显著减少IO次数