程序中的某条指令一旦执行,不久后该指令可能再次执行; 某数据被访问过,不久后该数据可能再次被访问。 产生时间局部性的典型原因是程序中存在大量循环操作 时间局部性 😑 一旦程序访问了某个存储单元,在不久后,其附近的存储单元也将被访问。即程序在一段时间内访问的地址,可能集中在一定的范围之内,因为指令通常是顺序存放、顺序执行的,数据也一般是以向量、数组、表等形式簇聚存储的。 局部性原理 一次性 传统存储管理方式的特征 🥞 驻留性 多次性 ★ 虚拟存储器的特征 ○ 对换性 概念 虚拟性 ★ 虚拟存储器的容量取决于地址空间的大小,而不是由实际的内存容量决定 一定容量的的内存和外存 页表机制(或段表机制),作为主要的数据结构。 虚拟内存技术的实现 〇 需要的支持 中断机构,当用户程序要访问的部门尚未调入内存时,产生中断 地址变换机构,逻辑地址到物理地址的变换 页号 物理块号 状态位 P 访问字段 A 修改位 M 外存地址 ·状态位 P: 用于指示该页是否己调入内存 ·访问字段 A: 用于记录本页在一段时间内被访问的次数,或记录本页最近己有多长时间未被访问 ·修改位 M: 标识该页在调入内存后是否被修改过 而夷机制 · 修改位 M: 标识该贝在铜入内存后是否被修改过 · 外存地址:用于指出该页在外存上的地址,通常是物理块号 在请求分页系统中,每当所要访问的页面不在内存时,便产生一个缺页中断,请求操作系统 将所缺的页调入内存 缺页中断机构 ◎ 在指令执行期间产生和处理中断信号,而非一条指令执行完后,属于内部中断。 一条指令在执行期间,可能产生多次缺页中断, 程序请求访问一页 开始 缺页中断处理 保留CPU现场 CPU检索快表 ★请求分页管理方式 该页被修改否? 操作系统命令CPU从外存读缺页 地址变换结束 地址变换机构 图 3-24 请求分页中的地址变换过程 访问页面 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1 物理块1 7 7 7 2 2 2 2 2 2 7 访问页面 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1 物理块1 7 7 7 2 2 2 4 4 4 0 0 0 0 7 7 7 优先淘汰最早进入内存的页面,亦即在内存中驻留时间最久的页面。 该算法实现简单,只需把调入内存的页面根据先后次序驻接成队列。设置一个指针总指向最早的页面。 但该算法与进程实际运行时的规律不适应,因为在进程中,有的页面经常被访问。 虚拟内存管理 先进先出页面置换算法【FIFO】 ⊙ FIFO算法还会产生当所分配的物理块数增大而页故障数不减反增的异常现象(Belady异常) 访问页面 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1 物理块1 7 7 7 2 2 2 4 4 4 4 0 1 1 1 1 物理块2 0 0 0 0 0 0 0 0 3 3 3 3 0 0 0 物理块3 1 1 1 3 3 2 2 2 2 2 2 7 缺页否 √ √ √ √ √ √ √ √ √ √ √ 该算法为每个贝面收置一个分时子段、米记录贝面自工次被访问以来所经历的时间,海汰贝面的选择现有贝面中值最大的予以海汰。 简单的CLOCK算法是给每一帧关联一个附加位、称为使用位。 当某一页首次装入主存时,该帧的使用位设置为1;当该页随后再被访问到时,它的使用位也被置为1。 对于页替换算法,用于替换的候选帧集合看做一个循环缓冲区,并且有一个指针与之相关联。 当某一页被替换时,该指针被设置成指向缓冲区中的下一帧。 当需要替换一页时,操作系统打描缓冲区,以直找使用位被置为0的一帧。 每当遇到一个使用位为1的帧时,操作系统就将该位重新置为0; 如果在这个过程开始时,缓冲区中所有帧的使用位均为0,则选择遇到的第一个帧替换; 如果所有帧的使用位均为1,则指针在缓冲区中完整地循环一周,把所有使用位都置为0,并且停留在最初的位置上,替换该帧中的页。 ★ 页面置换算法 CLOCK算法的性能比较接近LRU,而通过增加使用的位数目,可以使得CLOCK算法更加高效。 在使用位的基础上再增加一个修改位,则得到改进型的CLOCK置换算法。这样,每一帧都处于以下四种情况之一: 最近未被访问,也未被修改(u=0, m=0)。 最近被访问,但未被修改(u=0, m=0)。 最近未被访问,但被修改(u=1, m=0)。 CLOCK时钟置换算法【NRU】 最近被访问,被修改(u=1, m=1)。 算法执行如下操作步骤: 如果第2)步失败,指针将回到它的最初位置,并且集合中所有帧的使用位均为0。重复第1步,并且如果有必要,重复第2步。这样将可以找到供替换的帧。 给一个进程分配的物理页框的集合 它为每个进程分配一定数目的物理块,在整个运行期间都不改变。 若进程在运行中发生缺页,则只能从该进程在内存中的页面中选出一页换出,然后再调入需要的 页面。 ★ 驻留集 固定分配局部置换 🙃 为系统中的每个进程分配一定数目的物理块,操作系统自身也保持一个空阀物理块队列。 当某进程发生缺页时,系统从空阀物理块队列中取出一个物理块分配给该进程,并将欲调入的页装入其中。 驻留集大小 ⊙ ★ 三种策略 ⊙ 可变分配全局置换 ⊙ 它为每个进程分配一定数目的物理块,当某进程发生缺页时,只允许从该进程在内存的页面中选出一页换出可变分配局部置换 

动态变换,频繁缺页,分配物理块,缺页率低,增加物理块 将预计在不久后便会被访问的页面预先调入内存 主要用于进程的首次调入,由程序员指出应先调入哪些页 ★页面分配策略 调入页面的时机 请求调页策略 ② 进程在运行中需要访问的页面不再内存而提出请求,由系统将所需页面调入内存 对换区通常是采用连续分配方式,而文件区采用离散分配方式 对换区的磁盘 I/O 速度比文件区的更快。 系统拥有足够的对换空间 ○ 可以全部从对换区调入所需页面,以提高调页速度。在进程运行前,需将与该进程有关的文件从文件区复制到对换区。 凡不会被修改的文件都直接从文件区调入,而当换出这些回页面时, 由于它们未被修改而不必再将它们换出。 系统缺少足够的对换区空间 ○ 但对于那些可能被修改的部分,在将它们换出时必须调到对换区, 以后需要时再从对换区调入(这是因为读的速度比写的速度快)。 从何处调入页面 🕣 UNIX方式 在某段时间间隔内,进程要访问的页面集合。 ★工作集 频繁的页面调度行为 ★抖动