五.存储管理

A．一个非常重要的概念：地址空间

地址空间：最通俗的理解方式是物理空间的抽象表示。将物理空间映射到抽象空间，使操作系统便于对存储内容进行管理。每个进程都有自己的地址空间，进程通过地址空间对实际的物理内存进行操作。

当进程读取地址空间时，有两个概念：基址寄存器和界限寄存器。基址寄存器存储的是该进程拥有的地址空间的首地址，界限寄存器存储着该地址空间的最大地址在哪里。如果程序提供的地址加上基址寄存器的地址超出了界限寄存器的地址，则产生中断。

B．虚拟内存

1.基本思想是每一个程序都有自己的地址空间，这个空间被分解为一个个的页面，这些页面被映射到物理地址中，当然不是所有的地址空间的地址都被映射，是正在运行中的被映射。

虚拟地址 = 虚拟页号 + 偏移量

映射的工作原理;

程序运行时，将虚拟地址发送给 MMU（内存管理单元），然后 MMU 将虚拟地址转换为物理地址。在映射过程中，将虚拟空间划分为以 4K 为单位的页面，物理空间也是按照 4K 为单位划分的，便于映射。若程序调用的虚拟地址不在映射范围，此时 CPU 对操作系统进行系统调用（即陷入操作系统中），操作系统将该地址写入一个没有使用的磁盘中的页框，返回一个映射。

页表用于记录映射的关系，可以认为是一本字典的目录，页号为字典对应的具体页数。

2.实际中页表庞大，并且每个进程均有页表。此时需要用到分页加速算法。转换检测缓冲区（TLB，或称快表）：将一部分常用的页表项存入一个特定的缓冲区中，这个缓冲区会直接将虚拟地址转换为物理地址，不需要经过 MMU。

3.针对大内存

有两种解决方法：多级页表和倒排页表

1）多级页表：就是将一本字典分解为多本字典，且第一本字典是总目录，如果查找第一本字典，第一本会告诉这个地址在第几本字典，然后查找第二本字典的目录，依次查找。

2）倒排表项：通常页表在虚拟地址端建立，但这样建立导致页表占据空间很大。这时将页表放在物理地址端，会减少空间的占用。

C．页面置换算法

当发生缺页中断时，操作系统需要将一个页面换出内存，再把另一个页面调

入内存。这种置换算法对于 web 服务器的影响很大。常用的几种页面置换算法：

一．最优页面置换算法：无法实现，必须让操作系统知道在缺页中断发生时如何得知下个页面的访问时间。

二．最近未使用页面置换算法（Not recently used, NRU）：设置 R 位为访问位，M 为修改位，通过这两个位，操作系统可以将所有的页面分为 4 类，0 类，未访问未修改，1 类，未访问已修改，2 类，以访问未修改，3 类，以访问已修改。这个算法在每次清除内存中的页面时，会首先清除一个为访问已修改的页面。

三．先进先出置换算法（FIFO）：队列式置换算法，先进入内存中的页表会被首先清除。

四．第二次机会置换算法：在 FIFO 中添加一步，检查页面的 R 位，若为 1，则将 R 置零，然后放到队列末尾，若 R 为 0，则清除该页面。

五．时钟页面置换算法：当发生缺页中断时，检查此时的指向的页面的 R 位，若 R=0，淘汰页面，R=1，清除 R 位，将指针往前移。

六．最近最少使用页面算法（，least recently used，LRU）：在中断发生时，置换未使用时间最长的页面。

七．工作集页面置换算法：

工作集：一个进程正在使用的所有页面的集合或者一个更容易实现的定义，一个进程在过去 10ms 内内存中所有页面的集合。在进程还未运行时，为了让 CPU 尽可能少的产生缺页中断，会预先调页。对每一个工作集进行页面的扫描。

颠簸：每次执行几条指令后程序就发生一次缺页中断的过程。

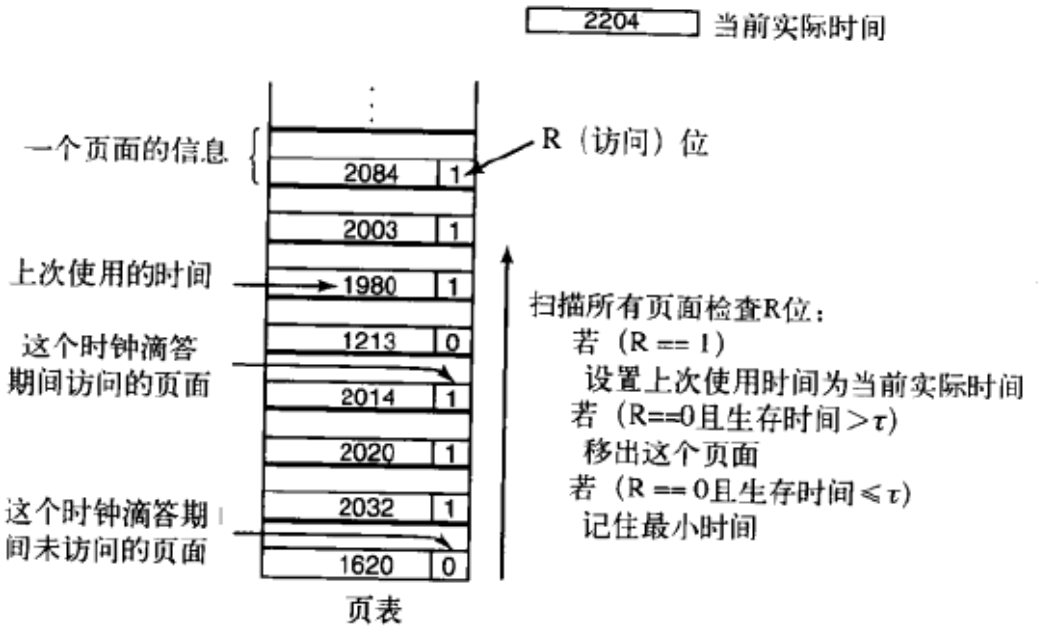

图3-20 工作集算法

八．工作集时钟页面置换算法（Wsclock 算法）：根据时钟去检查页面，类比时钟页面置换算法。

九．所有置换算法的一个总结：

算法	实现
最优	不可实现，但能作为一个基准
NRU	LRU的近似算法
FIFO	可能抛弃重要页面
第二次机会算法	比FIFO有较大改善
时钟算法	实际可用
LRU	很难实现，但性能很好
工作集算法	实现起来开销很大
工作集时钟算法	非常好的有效算法